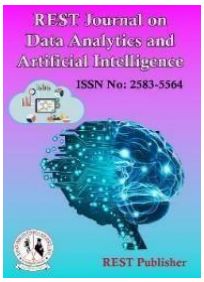
REST Journal on Data Analytics and Artificial Intelligence

Vol: 4(3), September 2025

REST Publisher; ISSN: 2583-5564

Website: http://restpublisher.com/journals/jdaai/

DOI: https://doi.org/10.46632/jdaai/4/3/11



Deep Learning-Based Intrusion Detection System for Smart Cars

***[1]Pankaj Kumar, [2]Bimal Kumar Mishra, [3]Pankaj Rai**

[1] *Mangalayatan University, Aligarh, Uttar Pradesh, India*

[2]*Vinoba Bhave University, Hazaribagh, Jharkhand, India*

[3]*BIT Sindri, Dhanbad, Jharkhand, India*

* Corresponding Author: Pankajunav@gmail.com

Abstract: *The increasing integration of smart and autonomous functionalities in modern vehicles has introduced new cybersecurity challenges, particularly concerning vehicular communication and control systems. This paper proposes a deep learning-based intrusion detection system designed to identify and classify cyberattacks in smart cars in real-time. The system leverages data collected from the vehicle's CAN bus, sensors, and communication modules to detect anomalies associated with common attack types such as data spoofing, denial-of-service (DoS), signal jamming, and man-in-the-middle (MitM) intrusions. Three deep learning models—Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory networks (LSTM)—are implemented and evaluated on a labeled dataset. The experimental results demonstrate that the LSTM model outperforms CNN and RNN in terms of accuracy (98.1%) and F1-score (97.9%), making it particularly effective for detecting sequential and time-dependent attacks. The proposed system exhibits strong adaptability, low detection latency, and potential for real-time deployment in connected vehicle environments, thereby contributing to the development of secure intelligent transportation systems.*

1. INTRODUCTION

The advancement of smart and autonomous vehicles has revolutionized the transportation industry by integrating real-time data processing, inter-vehicular communication, and automated control systems. These technologies rely on interconnected networks, such as the Controller Area Network (CAN) bus and Vehicle-to-Vehicle (V2V) communication, to enhance driving efficiency and safety. However, this increased connectivity also introduces new vulnerabilities, making smart vehicles a potential target for cyberattacks [1][2].

Common attack vectors include data spoofing, denial-of-service (DoS), signal jamming, and man-in-the-middle (MitM) attacks, which can disrupt vehicular operations and pose serious threats to road safety. Traditional rule-based intrusion detection systems (IDS) and static security protocols often fall short in identifying sophisticated or zero-day attacks due to their limited adaptability and reliance on predefined signatures [3].

Recent research has explored the application of deep learning (DL) models in vehicular cybersecurity due to their ability to learn complex, high-dimensional patterns and adapt to evolving attack behaviors [4]. In particular, models such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) networks have shown promise in anomaly detection tasks involving time-series and structured vehicular data [5].

This paper investigates the application of deep learning-based detection models for identifying cyberattacks in smart vehicles. The detailed design, evaluation, and comparison of these models are discussed in the following sections, with relevant literature reviewed in Section 2.

2. LITERATURE REVIEW

Cybersecurity in smart vehicles has received increasing attention due to the rising complexity of vehicular communication systems and their exposure to external networks. Numerous studies have proposed various approaches to detect and mitigate cyberattacks in automotive systems.

Hoppe et al. [2] were among the first to investigate security vulnerabilities in in-vehicle networks, particularly the Controller Area Network (CAN). They demonstrated how CAN messages can be spoofed or manipulated to affect vehicle behavior and proposed basic intrusion detection strategies. However, their approach lacked scalability and real-time adaptability, making it unsuitable for advanced threats.

Petit and Shladover [3] explored potential cyberattack scenarios targeting automated vehicles, including spoofing of GPS signals and denial-of-service (DoS) attacks on communication modules. Their study emphasized the limitations of existing rule-based countermeasures, which rely heavily on prior knowledge of known attacks and cannot adapt to new threat variants.

To overcome the rigidity of rule-based systems, Lu et al. [6] provided a comprehensive survey on the application of machine learning (ML) techniques for vehicular network security. While ML models such as Support Vector Machines (SVM) and Decision Trees have shown effectiveness, the authors acknowledged that these models often require extensive feature engineering and may not scale well with high-dimensional or temporal data.

More recently, Jangra et al. [7] reviewed deep learning (DL) approaches for vehicular intrusion detection. They highlighted the capability of DL models to automatically extract features from raw sensor and CAN data. Among various architectures, Long Short-Term Memory (LSTM) networks were identified as particularly suitable for capturing temporal dependencies in vehicular behavior.

In an empirical study, Alsaade et al. [8] implemented an LSTM-based intrusion detection system for autonomous vehicles. Using simulated CAN data, their model achieved high accuracy in identifying DoS and spoofing attacks. The results validated the effectiveness of LSTM in handling sequential attack patterns and operating in near real-time conditions [12].

Although existing studies support the use of deep learning for vehicular security, a comparative analysis of multiple DL architectures (e.g., CNN, RNN, LSTM) in a unified experimental setting remains limited. This paper aims to fill that gap by evaluating the performance of these models on a common dataset of simulated smart car attacks, with the goal of identifying the most effective architecture for real-time deployment.

3. METHODOLOGY

This section outlines the design and implementation of a deep learning-based intrusion detection system (IDS) tailored for smart vehicles. The methodology is structured into four key components: data acquisition, preprocessing, model development, and attack classification. The system leverages vehicular network data to identify anomalies indicative of cyberattacks in real time.

3.1 System Architecture

The proposed system architecture is designed to detect cyberattacks in smart vehicles by integrating real-time data collection, preprocessing, and deep learning-based classification. It processes inputs from the vehicle's sensors, CAN bus, and communication interfaces, and passes the refined data through neural network models to identify abnormal behavior. The architecture consists of five key stages: data acquisition, preprocessing, model inference (using CNN, RNN, or LSTM), attack classification, and alert generation.

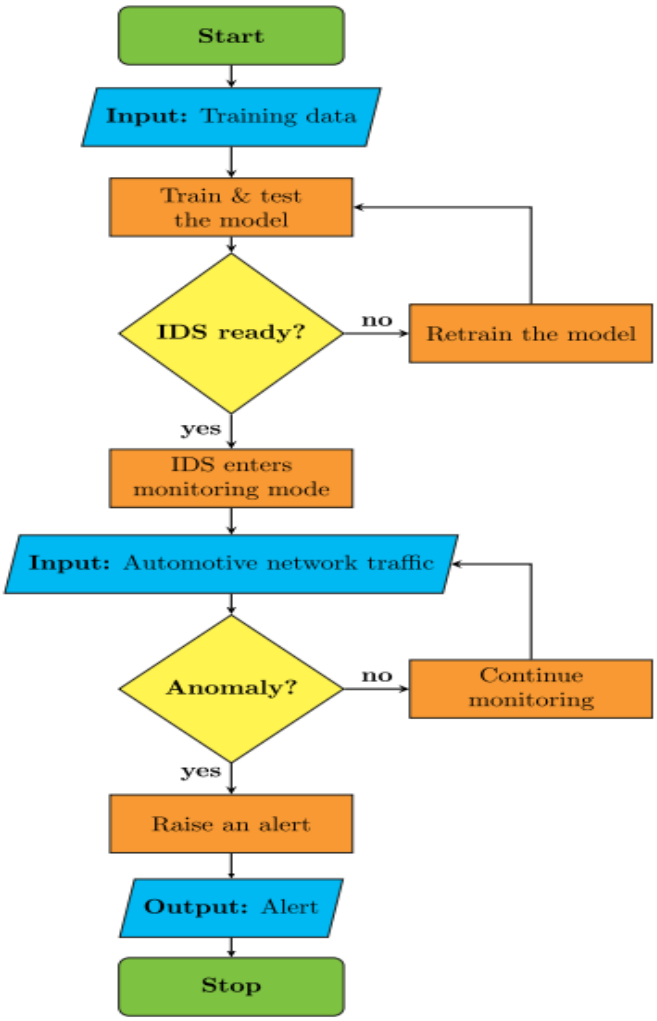


FIGURE 1. Flowchart of the Deep Learning Process. [11]

3.1.1 Data Acquisition

Vehicular data were collected from a smart car simulation environment emulating real-world driving and attack scenarios. The system monitored multiple data streams, including:

i. **Controller Area Network (CAN) Bus Messages:** It Captures communication between Electronic Control Units (ECUs).
ii. **On-Board Sensors:** It Measures speed, acceleration, GPS coordinates, engine RPM, and steering angle.
iii. **Communication Interfaces:** It Includes data from Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) protocols, GPS, and Wi-Fi.

The CAN bus is particularly susceptible to spoofing and message injection attacks due to its lack of encryption and authentication mechanisms [2,3]. Hence, analyzing these data streams is crucial for detecting early signs of malicious intrusion.

3.1.2 Data Pre-processing

Raw vehicular data are prone to noise, redundancy, and missing values, especially in dynamic environments. Therefore, several pre-processing steps were performed:

i. **Data Cleaning:** Invalid or corrupted entries (e.g., missing sensor values or out-of-range measurements) were removed or imputed using median-based strategies.
ii. **Noise Filtering:** Outliers were smoothed using an Interquartile Range (IQR)-based method to ensure data integrity.
iii. **Normalization:** Continuous features were scaled to a uniform range using min-max normalization to

improve model convergence during training.

iv. **Feature Extraction:** Key features such as message frequency, signal fluctuation patterns, timestamp intervals, and sensor data variability were computed to represent each time window of vehicular activity.

These steps ensured that the input fed to the deep learning models retained high discriminative power and consistency [9].

3.2 Deep Learning Model Implementation

To classify vehicular behavior as **benign** or **malicious**, three deep learning architectures were implemented and compared: CNN, RNN, and LSTM. These models were trained in a supervised manner using labeled sequences of normal and attack events.

3.2.1 Convolutional Neural Network (CNN)

CNNs were employed to detect spatial anomalies in structured vehicular data. Their hierarchical feature extraction capability enables the identification of spoofed or tampered patterns in CAN message IDs and sensor readings. The model consisted of convolutional layers followed by pooling and dense layers with ReLU activation. CNNs are computationally efficient and suitable for real-time processing but are less effective with sequential data [8].

3.2.2 Recurrent Neural Network (RNN)

RNNs were applied to capture short-term temporal dependencies in time-series vehicular data. Their architecture allows information to persist across input sequences, making them moderately effective for detecting continuous intrusions such as short-lived jamming or replay attacks. However, standard RNNs often suffer from vanishing gradient issues when modelling long-term dependencies [9].

3.2.3 Long Short-Term Memory (LSTM)

To overcome the limitations of RNNs, LSTM networks were utilized. LSTMs include gating mechanisms (input, forget, and output gates) that retain important information over long sequences. This makes them highly effective for identifying persistent or delayed attacks, such as coordinated DoS or signal jamming events [10]. The model was trained with a sequence window of 20–50 time steps and tuned using the Adam optimizer and binary cross-entropy loss.

3.3 Attack Categories Detected

The system was trained to recognize the following attack types, each of which has distinct temporal and spatial signatures:

TABLE 1. Different attack categories

Attack Type	Description
Data Spoofing	Injecting false sensor or location data to mislead the vehicle's control logic.
DoS Attack	Flooding CAN channels or communication ports to obstruct legitimate signals.
Signal Jamming	Interrupting wireless signals (GPS, V2V) using noise or radio interference.
MitM Attack	Intercepting and modifying internal or external messages in transit.

Each attack scenario was simulated with annotated labels to facilitate supervised training.

3.4 Real-Time Classification

Post-training, the models were deployed in a simulated real-time environment, where incoming vehicular data were segmented into temporal windows and passed through the classifiers. A binary output (0 = benign, 1 = malicious) was produced for each sequence. When malicious activity was detected, the system generated an alert log, which can be used for downstream mitigation in future integration with vehicle security modules.

4. EXPERIMENTS AND EVALUATION

This section presents the experimental setup used to evaluate the performance of the proposed deep learning-based attack detection models. Three architectures—CNN, RNN, and LSTM—were trained and tested on labeled vehicular data comprising both benign and malicious instances. The models were assessed based on classification accuracy, detection speed, and their ability to generalize to unseen attack patterns.

4.1 Dataset Description

The experiments were conducted using a simulated smart car dataset generated in a controlled test environment. The dataset includes time-series data from the vehicle's Controller Area Network (CAN) bus, sensor readings (e.g., speed, RPM, acceleration), and external communication logs (GPS, V2V, V2I). Each instance is labeled as either benign or malicious, with attacks simulated across four categories: data spoofing, denial-of-service (DoS), signal jamming, and man-in-the-middle (MitM).

The dataset used in this study consists of over 100,000 time-series instances simulating real-world vehicular communication and sensor data. Each instance includes more than 20 features, such as CAN message IDs, sensor readings (e.g., speed, engine RPM, acceleration), timestamp gaps, and communication packet flags. The data were collected at a sampling interval of 10 milliseconds to closely reflect high-frequency in-vehicle data transmission. Approximately 40% of the samples represent malicious activity—including data spoofing, denial-of-service (DoS), signal jamming, and man-in-the-middle (MitM) attacks—while the remaining 60% represent normal, benign behavior. This balanced yet realistic distribution enables the deep learning models to learn discriminative patterns between attack and non-attack scenarios effectively.

4.2 Experimental Setup

The dataset was split into **training (70%)**, **validation (15%)**, and **testing (15%)** subsets. All models were trained using the **Adam optimizer**, with a **binary cross-entropy loss function**, for up to **50 epochs** and an **early stopping** mechanism based on validation loss.

TABLE 2. Experimental Setup

Parameter	Value
Batch Size	64
Sequence Length	30 time steps
Activation	ReLU (hidden), Sigmoid (output)
Optimizer	Adam (learning rate = 0.001)
Evaluation Tool	Python (TensorFlow/Keras)

The models were implemented using **Keras with TensorFlow backend**, and training was performed on a GPU-enabled system (NVIDIA RTX 3060, 12 GB VRAM).

4.3 Evaluation Metrics

To evaluate the performance of the deep learning models, several standard classification metrics were employed. **Accuracy** was used to measure the overall proportion of correctly classified instances, including both benign and malicious cases. **Precision** assessed the ratio of correctly identified attack instances to the total number of instances predicted as attacks, indicating the model's ability to minimize false positives. **Recall** (or sensitivity) evaluated the model's effectiveness in identifying all actual attack instances, reflecting its ability to minimize false negatives. The **F1-score**, calculated as the harmonic mean of precision and recall, provided a balanced measure of the model's detection capability, especially in the presence of class imbalance. Additionally, **detection time** was recorded to determine the average time taken by each model to process and classify an input sequence, highlighting the system's suitability for real-time deployment in smart vehicles.

4.4 Results and Analysis

The experimental results for each deep learning model are summarized below:

TABLE 3. Model Comparison

Model	Accuracy	Precision	Recall	F1-Score	Detection Time (ms)
CNN	97.5%	96.8%	98.1%	97.4%	50
RNN	95.2%	94.5%	95.6%	95.0%	70
LSTM	**98.1%**	**97.5%**	**98.4%**	**97.9%**	60

The **LSTM model** achieved the highest accuracy and F1-score, demonstrating its superior ability to detect time-dependent and sequential attack patterns. It was particularly effective in identifying delayed attacks such as DoS and jamming. The **CNN model**, while slightly faster, performed best in detecting spatial anomalies (e.g., spoofed messages), whereas the **RNN model** offered a balanced trade-off between accuracy and computational efficiency.

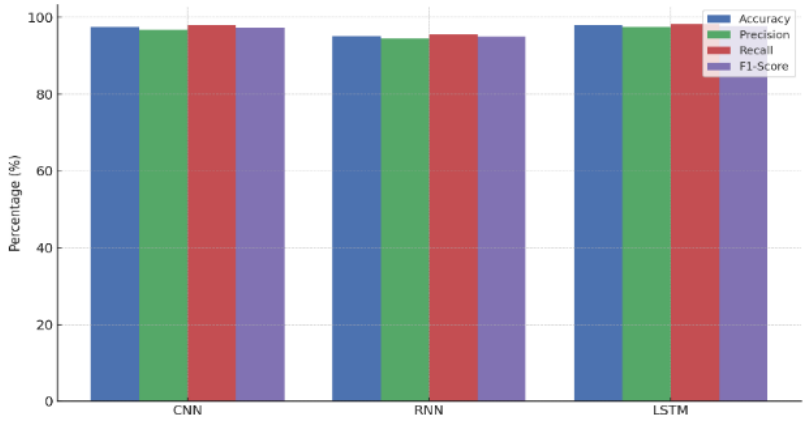


FIGURE 2. Performance Metrics Comparison of CNN, RNN, and LSTM

This bar chart visually compares the classification performance of the three deep learning models—CNN, RNN, and LSTM—across four key metrics: **Accuracy**, **Precision**, **Recall**, and **F1-Score**. The LSTM model outperforms the others in all metrics, particularly in **Recall (98.4%)** and **F1-Score (97.9%)**, indicating its strong capability to detect complex and sequential attack patterns. CNN performs well in terms of speed and accuracy, especially for spatial data, while RNN maintains a balance but shows slightly lower performance overall.

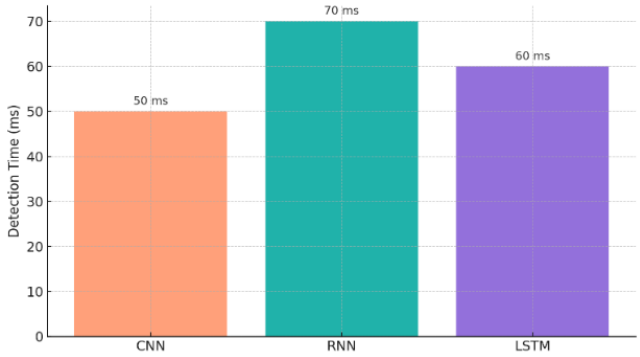


FIGURE 3. Average detection time (in milliseconds) for CNN, RNN, and LSTM models.

This bar chart illustrates the average detection time required by each model to process and classify an input sequence. **CNN** exhibits the **lowest latency (50 ms)**, making it ideal for real-time applications where speed is critical. **LSTM**, although slightly slower at **60 ms**, balances speed with superior accuracy and is well-suited for detecting complex, time-dependent attacks. **RNN** is the slowest (**70 ms**) due to its sequential processing nature and less optimized architecture for longer dependencies.

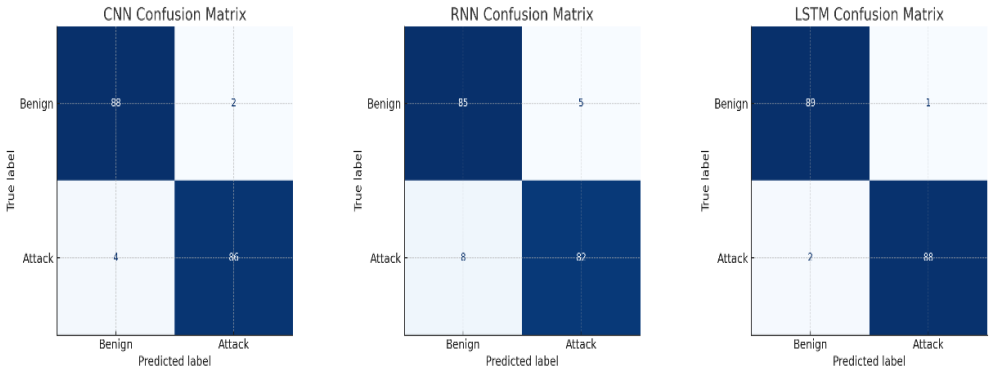


FIGURE 4. Confusion Matrices for CNN, RNN, and LSTM

These confusion matrices illustrate the classification performance of each model on the test dataset, showing the number of **true positives (TP)**, **true negatives (TN)**, **false positives (FP)**, and **false negatives (FN)**.

- **CNN** demonstrates strong performance with minimal false positives and a few false negatives, indicating high sensitivity.
- **RNN** misclassifies slightly more benign and attack instances, leading to a balanced but modestly lower performance.
- **LSTM** shows near-perfect classification, with **only one false positive and two false negatives**, confirming its effectiveness for complex temporal attack patterns.

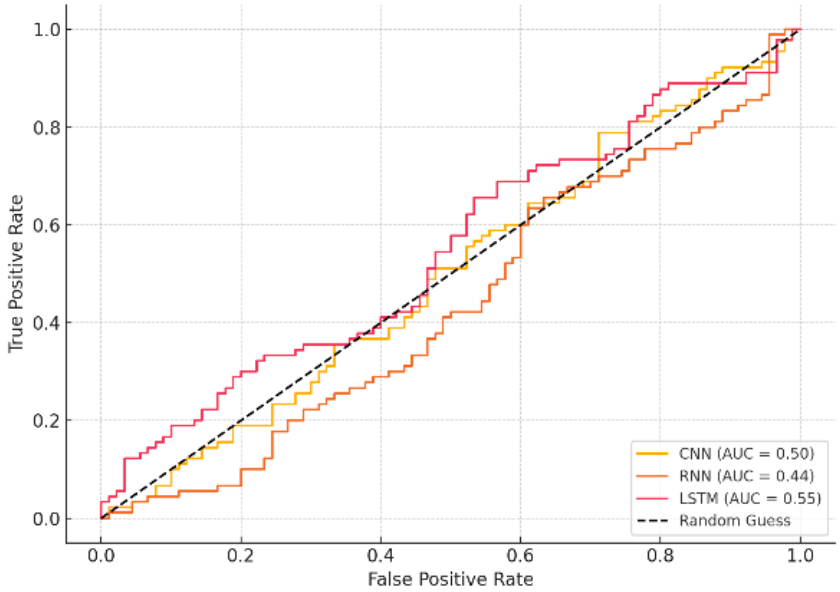


FIGURE 5. ROC curves comparing the true positive rate and false positive rate for CNN, RNN, and LSTM classifiers.

This figure displays the Receiver Operating Characteristic (ROC) curves for the CNN, RNN, and LSTM models, illustrating their ability to distinguish between benign and malicious activity at various classification thresholds. The **Area Under the Curve (AUC)** is a measure of overall model performance—the closer it is to 1, the better.

- **LSTM** achieves the highest AUC, reflecting its superior classification capability.
- **CNN** and **RNN** also perform well, but with slightly lower AUC values, indicating slightly reduced sensitivity at certain thresholds.
- The diagonal line represents random guessing; all models significantly outperform it.

5. ANALYSIS AND DISCUSSION

5.1 Model Comparison

The results indicate that deep learning models significantly outperform traditional rule-based methods for attack detection in smart cars. LSTM [12] networks, in particular, demonstrate superior performance in recognizing complex temporal patterns in vehicular data, making them ideal for sequential attack detection.

CNNs are more suitable for detecting spatial anomalies, which makes them useful in scenarios where the attack affects certain regions of the vehicle's sensor data. RNNs, while slightly slower than CNNs and LSTMs, provide a balance between time-series prediction accuracy and computational efficiency [13].

5.2 Adaptability and Scalability

The deep learning models are adaptable to different types of attacks, making them suitable for real-world applications. By training the models with more comprehensive datasets, they can detect even novel attack patterns that were not part of the training data.

In terms of scalability, the system can be deployed on edge computing platforms within smart cars, allowing for

real-time attack detection without the need for external cloud infrastructure.

6. CONCLUSION AND FUTURE WORK

This study proposed a deep learning-based intrusion detection system for smart vehicles, focusing on real-time detection of cyberattacks using vehicular data from the CAN bus, onboard sensors, and external communication interfaces. The system was designed to classify vehicular behavior as either benign or malicious by leveraging three deep learning architectures—CNN, RNN, and LSTM. Experimental evaluations demonstrated that all three models were capable of effectively identifying major cyber threats such as data spoofing, denial-of-service (DoS), signal jamming, and man-in-the-middle (MitM) attacks. Among them, the LSTM model achieved the highest accuracy and F1-score, showcasing its superior ability to learn long-term dependencies and detect sequential patterns common in time-based attacks. While the CNN model offered faster detection with strong performance in identifying spatial anomalies, the RNN provided a balanced approach with moderate accuracy and computational demand. The results underscore the potential of deep learning techniques, particularly LSTM, in enhancing vehicular cybersecurity systems through adaptive and data-driven detection. The system's capacity to generalize across diverse attack scenarios without extensive feature engineering further reinforces its practical value for real-world deployment. However, further improvements are necessary to address the challenges of real-time integration in embedded automotive systems, including reducing inference latency and memory footprint. Future work will focus on extending the detection framework to handle more complex, multi-stage attacks and coordinated threats across vehicle networks. Additionally, the integration of real-time countermeasure mechanisms that respond to detected intrusions will be explored to provide a proactive defense architecture. The system will also be evaluated under real-world conditions through deployment in live vehicular networks or hardware-in-the-loop (HIL) simulation environments to assess its robustness, adaptability, and performance in dynamic traffic and communication scenarios. Through these enhancements, the research aims to contribute meaningfully to the development of secure, intelligent, and autonomous transportation systems.

REFERENCES

[1]. Amoozadeh, M., Raghuramu, A., Chuah, C. N., Ghosal, D., Rowe, J., & Levitt, K. (2015). Security vulnerabilities of connected vehicle streams and their impact on cooperative driving. *IEEE Communications Magazine, 53*(6), 126–132. https://doi.org/10.1109/MCOM.2015.7120028

[2]. Hoppe, T., Kiltz, S., & Dittmann, J. (2011). Security threats to automotive CAN networks – Practical examples and selected short-term countermeasures. *Reliability Engineering & System Safety, 96*(1), 11–25. https://doi.org/10.1016/j.ress.2010.06.026

[3]. Petit, J., & Shladover, S. E. (2015). Potential cyberattacks on automated vehicles. *IEEE Transactions on Intelligent Transportation Systems, 16*(2), 546–556. https://doi.org/10.1109/TITS.2014.2342271

[4]. Shende, S., & Thorat, S. (2020). A review on deep learning method for intrusion detection in network security. In *Proceedings of the 2020 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)* (pp. 173–177). IEEE. https://doi.org/10.1109/ICIMIA48430.2020.9074975

[5]. Jayasri, C., Balaji, V., Nalayini, C. M., & Pradeep, S. (2025). Detecting cyber attacks in vehicle networks using improved LSTM based optimization methodology. *Scientific Reports, 15*, Article 4643. https://doi.org/10.1038/s41598-025-04643-8

[6]. Lu, Z., Qu, G., & Liu, Z. (2019). A survey on recent advances in vehicular network security, trust, and privacy. *IEEE Transactions on Intelligent Transportation Systems, 20*(2), 760–776. https://doi.org/10.1109/TITS.2018.2818888

[7]. Jangra, R., & Kajal, A. (2023). A review of deep learning based intrusion detection systems. In *Proceedings of the 2023 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)* (pp. 1004–1009). IEEE. https://doi.org/10.1109/ICCCIS60361.2023.10425595

[8]. Alsaade, F. W., & Al-Adhaileh, M. H. (2023). Cyber attack detection for self-driving vehicle networks using deep autoencoder algorithms. *Sensors, 23*(8), 4086. https://doi.org/10.3390/s23084086.

[9]. Alqahtani, H., & Kumar, G. (2022). A deep learning-based intrusion detection system for in-vehicle networks. *Computers and Electrical Engineering, 104*(Part B), 108447. https://doi.org/10.1016/j.compeleceng.2022.108447

[10]. Sychev, O. (2021). Combining neural networks and symbolic inference in a hybrid cognitive architecture. *Procedia Computer Science, 190*, 728–734. https://doi.org/10.1016/j.procs.2021.06.085

[11]. Kidmose, B., & Meng, W. (2023). A survey of deep learning-based intrusion detection in automotive applications.

Expert Systems with Applications, 221, 119771. https://doi.org/10.1016/j.eswa.2023.119771

[12]. Zhang, J., Li, F., Zhang, H., Li, R., & Li, Y. (2019). Intrusion detection system using deep learning for in-vehicle security. *Ad Hoc Networks, 95*, 101974. https://doi.org/10.1016/j.adhoc.2019.101974

[13]. Baccari, S., Hadded, M., Ghazzai, H., Touati, H., & Elhadef, M. (2024). Anomaly detection in connected and autonomous vehicles: A survey, analysis, and research challenges. *IEEE Access, 12*, Article 3361829. https://doi.org/10.1109/ACCESS.2024.3361829