Nujud Al-aql[*1],
Dr. Abdulaziz Al-Shammari [*1]

Efficient Intrusion Detection in CAN Bus Networks

Journal of Electrical Systems

Abstract: Electric Vehicles (EVs) rely on electric motors powered by complex battery systems to ensure efficient propulsion. A critical element in their performance is the Controller Area Network (CAN) protocol, widely adopted in the automotive industry for seamless communication among vehicle components, including Electronic Control Units (ECUs). Originally designed with minimal security considerations, the CAN protocol exposes modern connected vehicles to various cyber threats. Attacks such as Denial of Service (DoS), Fuzzy, and Impersonation pose significant risks to vehicle safety and operational integrity, highlighting the urgent need for robust Intrusion Detection Systems (IDS) tailored to CAN networks. In this study, an advanced hybrid detection model, named RL-RF Guard, is proposed, integrating Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) units, and Random Forest (RF) classifiers to enhance intrusion detection in EV CAN systems. The approach begins with an RNN extracting initial features from CAN traffic, which are further refined by the LSTM network to capture temporal attack patterns. Finally, a Random Forest classifier leverages these deep features to achieve more precise classification between normal and malicious CAN messages. Evaluation on a comprehensive dataset reveals that the RL-RF Guard model attains an accuracy of 98%, significantly enhancing detection capability compared to prior models. This research presents a novel and effective strategy for vehicular network security, combining the temporal learning strengths of RNN and LSTM with the robust classification capabilities of RF The findings underscore the potential of such hybrid architectures to form the backbone of next-generation IDS solutions, ensuring the resilience and trustworthiness of connected vehicles against evolving cyber threats.

Keywords: Intrusion detection system (IDS); controller area network (CAN) bus; deep learning (DL); Recurrent Neural Networks (RNN); Long Short-Term Memory (LSTM) networks; Denial of Service (DoS); Electronic Control Units (ECUs), Random Forest (RF).

I. INTRODUCTION

The rapid development of technology for EVs has redefined the automotive industry, which has produced new possibilities that are productive and environmentally sustainable. The modern EV relies on complex systems that manage electric motors, batteries, and other attachments so as to enable such features as autonomous driving, real-time diagnosis, and enhanced user experience. Under these systems, the Controller Area Network (CAN) protocol is a stable data communication medium for passing necessary critical information between ECUs. As control systems for many aspects of a vehicle, such as the engine, brakes, and infotainment, ECUs make the vehicle work smoothly while lowering the amount of noise, vibration, and energy consumed. Connecting a variety of ECUs using CAN bus has, to a great measure, enhanced the vehicle functions, making automobiles not only smarter but also more responsive.

However, the increased level of connection and complexity in EVs has triggered serious cybersecurity issues. Although the CAN protocol was initially designed in the 80s, it does not offer basic protection features like authentication and encryption. Due to that, the CAN protocol becomes vulnerable to cyberattacks, which potentially might compromise vehicle safety and performance. For instance, the attackers exploit what is insecure, such as OBD ports and USB interfaces, Bluetooth, Wi-Fi, and 5G connectivity, to introduce destructive signals into the CAN network. Consequently, such types of attacks may be able to tamper with critical vehicle operations such as steering, braking, and engine management, posing significant threats to all occupants of the vehicle. The Denial of Service (DoS) attacks are common and consist of flooding a network with requests to prevent orderly communication. Fuzzy attacks are the insertion of random data to cause malfunctioning processes, and impersonation attacks correspond to when an attacker gets in by imitating a real ECU.

1 College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU) , Riyadh 11432, Saudi Arabia , e-mail:435030595@sm.imamu.edu.sa

The presence of sophisticated vehicular cyberattacks has been confirmed in actual cases reported in the world. Researchers confirm that attackers can enter a vehicle's systems from a distance by exploiting key fob or infotainment system weaknesses, thereby putting critical controls and confidential data at risk. These examples make clear the need for effective IDS solutions to defend EV networks. Security strategies that depend on access control or basic encryption seldom match the evolving and constrained requirements of CAN networks. In addition, the processing constraints of ECUs hinder the deployment of robust security protocols without introducing performance issues. To address those problems, for this research, we proceed with an approach to employing deep learning techniques, namely RNN and LSTM networks, to develop a smart IDS for EV CAN systems. As it is known that deep learning is useful in identifying anomalies and patterns from massive and complex sets of data, its application is an appropriate method for identifying cyber threats when they occur. The hybrid model that is offered makes use of the strengths of RNNs for sequential data and LSTMs for long-term dependencies to obtain superior detection accuracy. The analysis of CAN messages currently carried out by this model targets to provide efficient cybersecurity to connected vehicles by deterring intrusions and maintaining their safety and reliability.

The purpose of this research is to narrow the gap between the EVs' rapidly growing connectivity and the drawbacks of the current security allowed. This research attempts to benefit from the deep learning power to develop an IDS that adaptively protects the automotive domain from evolving attacks. Such results should increase security standards, sustain public confidence in electric vehicle technology, and eventually deliver safer and more resistant vehicles in a connected space

II. BACKGROUND

To guarantee optimal performance, safety, and efficiency, contemporary automotive systems, mainly in EVs, depend on advanced communications and control approaches. These systems rely fundamentally on the Controller Area Network (CAN) protocol, as well as on Electronic Control Units (ECUs), which together serve as the essential foundation for in-vehicle communication. Consequently, the growing connectedness of vehicles has brought to light several weaknesses, which now make them attractive targets for cyberattacks. Scholars have proposed using deep learning to solve these security challenges by greatly enhancing the robustness of intrusion detection methods. In this section, the CAN protocol, ECUs, the nature of cyberattacks on EVs, and how deep learning supports vehicle cybersecurity are described in detail.

- **CAN Protocol**

In the 1980's, in the Robert Bosch GmbH, CAN protocol has been developed, which has become a good and widespread standard for real-time data transfer for the automobile and industrial systems. Designed to save communications between vehicles, the protocol of CAN protocol was later a main contributor to up-to-date technologies of the automobile, due to its reliability and effective capabilities. Unlike network designs built around a host, CAN operates as a message-based decentralized protocol such that several ECUs can interact effectively without relying on any one for control. Consequently, this architecture provides a platform on which the rapid and dependable exchange of data among vital vehicle systems such as the engine, brake, and transmission controls becomes possible. Differential signaling by CAN considerably adds to its ability to resist electromagnetic noise, a normal issue in vehicles. The transmitted data is organized into frames, each marked by a number indicating the priority of the contents of the data and payload of up to eight bytes. The resolution mechanism of CAN is a bit-wise arbitration, which gives priority to higher priority identifiers; their associated messages are sent first. Due to its fault-tolerant model and dynamic rate at which data is delivered from a few kilobits up to several megabits per second, CAN offers high flexibility in usage across system uses. With improvements to the CAN protocol, for which CAN Flexible Data-Rate (CAN FD) is an example, its data throughput and detection of errors have been enhanced, thus securing its place in automotive networking, industrial automation, and more. Even if the CAN protocol is strong in many aspects, it is not equipped with cybersecurity features from the design. Since there isn't encryption or authentication, anyone connected to the CAN bus can write or read messages without limits. This weakness has come to be a key problem since the connectivity of cars via interfaces such as Wi-Fi, Bluetooth, and cellular communication increases the probability of unauthorized intrusion and malicious attacks.

- **Electronic Control Units**

  Electronic Control Units (ECUs) are customized microcomputer systems that provide the intellectual capabilities needed to control and enhance the performance, safety, and experience in contemporary vehicles. The modules in each ECU are a microcontroller or microprocessor, memory, input/output interfaces, and communication components, supporting the real-time processing of sensor data, algorithm execution, and actuator control. A typical function of an ECU is to control the delivery of fuel, affect the amount of braking force, and handle audio and navigation applications. Previously, vehicle control functioned using mechanical and hydraulic systems that struggled to handle behavior requiring sophisticated processing. The use of ECUs became a significant technological breakthrough, making it easier to control vehicle systems with great accuracy and introducing features like adaptive cruise control and lane-keeping support, as well as fully autonomous vehicles. Each modern vehicle generally contains several ECUs, with each specialized for functions including engine control, transmission settings, or the use of ADAS. Modular architecture simplifies the whole process of diagnosing problems, maintaining the vehicle, and updating software, and it even allows the integration of advanced machine learning methods to review driving behavior and support greater safety. Still, the growing number of ECUs within vehicles has led to a more complex architecture and introduced new security problems. Because the CAN bus is unsecured, ECUs sending messages over it cannot confirm the source or validate the content of those messages. This poses a threat because a malicious ECU might command other ECUs, which in turn may disrupt important vehicle operations. With the rise in the use of external interfaces, including OBD ports and wireless connections, unauthorized access becomes more likely, underscoring the requirement for robust cybersecurity measures to safeguard both vehicles and occupants.

- **Attack on EVs**

  The growing digitization of electric vehicles has made them an attractive target for cybercriminals looking to exploit flaws in their communication systems. The lack of security measures in the CAN protocol, including authentication and encryption for EV, makes EV vulnerable to a number of cyber-attacks capable of undermining the security and functionality, and data privacy. Attacks may be performed remotely on a wireless interface, such as Bluetooth, Wi-Fi, or cellular network, or physically by direct access to the OBD ports, USB drives, or other inputs on the manufacturing or restoration level. Some of the most common attack types include Denial of Service (DoS), which applies an excessive number of messages over the CAN bus in order to disrupt communication; Impersonation attacks, which impersonate legitimate ECUs, to send false commands; and the Fuzzy attacks involve injecting random data to induce unpredictable malfunctions. Such attacks can be critical and include hijacking critical systems such as braking, steering, manipulating the infotainment system, or stealing users' sensitive data. These vulnerabilities have real-world consequences, a fact that has been demonstrated by high-profile demonstrations by researchers whereby attackers can remotely control a vehicle's engine or brakes over the internet. The networking nature of modern vehicles further increases the effects of such attacks. For instance, a compromised infotainment system could be a way to access the CAN bus to manipulate the systems critical to the vehicle's operation. In the same manner, flaws in the key fob or mobile app for remote vehicle control may be entry points for bad actors. Not only are these threats dangerous for passengers, but negatively affect public trust in EV technology, which makes it clear that sophisticated security solutions are needed to protect connected vehicles from theoretical and practical threats.

- **Deep Learning for Detection**

  Deep learning has taken shape as a strong mechanism for dealing with the cybersecurity threats to EV CAN systems. Unlike those traditional security methods, which are mostly based on static security rules or lightweight encryptions, the Deep Learning models can be used to analyze large volumes of data for recognizing patterns or anomalies related to cyberattacks. Such models are especially appropriate for recording the double dangerously fleet momentariness of CAN communication, where even the slightest advancement in the delivery of messages, their punctuality, frequency, or order might indicate an intrusion. For CAN systems, Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks are specifically good at intrusion detection. RNNs are created to process sequential data, and, therefore, perfectly fit solving the task of detecting the temporal patterns of CAN messages. However, long-term dependencies are problematic with standard RNNs, such as vanishing gradients. LSTMs overcome this limitation because they are equipped with memory

cells that store information for long times, thus being able to detect advanced attacks that transform with time, for example, gradual data manipulation. Using deep learning models, these models can be trained to detect normal and malicious CAN messages by learning the normal behaviour of a vehicle's network. For instance, they can spot the differences in message intervals or detect such unexpected commands that do not comply with common patterns. In spite of its benefits, deep learning methods are not without their challenges; the need for incredibly large, high-quality data and the risk of adversarial attacks, where an attacker exploits data to avoid detection. Continuing studies will enhance the ruggedness and efficiency of these models and thereby make them a vital part of the next generations of intrusion detection systems for the protection of EV networks.

III. RELATED WORK

As cyber assaults on electric vehicles (EVs) become more widespread, there has been a flurry of research into making the Controller Area Network (CAN) protocol more secure, an important communication standard in automotive systems. Due to the fact that the CAN protocol doesn't have inherent security features such as authentication and encryption available, researchers have investigated different methods to generate efficient Intrusion Detection Systems (IDS). ML and DL methods have received much attention because they can extract detailed data patterns and discover live anomalies. This part reviews recent works that use ML and DL for the detection of intrusion, utilizing the overview of their methodologies, metrics of performance, and weaknesses.

- **Machine Learning for Detection on CAN Protocol**

There are many techniques of machine learning that have been broadly applied to intrusion detection in the CAN systems using statistical and pattern recognition functionality to recognize illicit practices. Such methods usually require the training of models on datasets that combine normal as well as attack-related CAN messages in order to correctly label anomalies.

Yang et al. proposed a lightweight ML framework for IDS in autonomous vehicles, detailing their approach for tree-based decision classifiers (such as Decision Trees (DT)). From using the HCRL car-hacking dataset, their model achieved an accuracy of 99.99% (recall: 99.99%, F1-score: 0.999). In the CICIDS2017 dataset, when the model is tested, it performed with very high accuracy, 99.72, and very high recall, 99.3, but with a slightly improved F1 score of 0.998. However, the study indicated that extra validation of the model on real-time datasets was also required to validate its strength in dynamic settings.

Alfardus and Rawat have performed a comparative study by using the HCRL Car-Hacking dataset, which assesses four algorithms. Support Vector Machine (SVM), Random Forest (RF), K-Nearest Neighbor (KNN), and a deep learning model. From the results, their achieved performance was near-perfect for SVM and RF, with the accuracy and recall, and F1-scores all 100% and 99.99%. KNN with a slightly lower performance scored an effective accuracy of 98.82% and a recall of 98.04%. The study also revealed the predictability of the ML in detecting attacks, with a line that real-world testing is still required for addressing practical challenges.

An IDS for CAN systems was suggested by Alshammari et al using the HCRL CAN intrusion detection dataset. They compared SVM and KNN, such that KNN gave a better performance than SVM delivering an accuracy of 96.9%, recall of 98.5%, and an F1-score of 93.5% over SVM's 96.4% accuracy, The study highlighted KNN's outstanding performance with handling the complexity of the dataset but added that both of the models could use optimization in resource-constrained ECUs.

D'Angelo, Castiglione, and Palmieri designed a cluster-based anomaly detection framework through the HCRL CAN Intrusion Detection dataset. Their approach employed two algorithms: a Cluster-based Learning Algorithm, in order to generate baseline patterns of CAN messages, and a Data-driven Anomaly Detection Algorithm with a runtime classification feature. This approach has demonstrated great accuracy of 99.98%, with precision at 99.86%, thus qualifying it for identifying lawful messages from malicious ones. Yet, the research did not test the framework's performance in adverse conditions, which may undermine its practicality.

Refat et al. explored a novel method that defined graph-based features from CAN messages beforehand and used ML methods upon them. On the HCRL Car-Hacking dataset, they tested SVM and KNN, SVM outperforming KNN with 99.67% accuracy, 97.23% recall, 98.04% F1-score, respectively, and 98.5. Though the graph-based transformation enhanced feature representation, it introduced computational overhead that may hamper real-time deployment.

The following table summarizes the performance of ML-based IDS for CAN systems:

Table 1 ML INTRUSION DETECTION

Authors	Type	Accuracy (%)	Recall (%)	F1-Score	Precision (%)	Dataset
Yang et al.	DT	99.99	99.99	0.999	-	HCRL Car-Hacking
Yang et al.	DT	99.72	99.3	0.998	-	CICIDS2017
Alfardus and Rawat	SVM	99.99	100	1.0	-	HCRL Car-Hacking
Alfardus and Rawat	RF	99.99	100	1.0	-	HCRL Car-Hacking
Alfardus and Rawat	KNN	98.82	98.04	1.0	-	HCRL Car-Hacking
Alshammari et al.	SVM	96.4	97.7	0.933	98.4	HCRL CAN Intrusion
Alshammari et al.	KNN	96.9	98.5	0.935	99.9	HCRL CAN Intrusion
D'Angelo, Castiglione, and Palmieri	Clustering	99.98	-	-	99.86	HCRL CAN Intrusion
Refat et al.	SVM	99.67	97.23	0.9804	99.03	HCRL Car-Hacking
Refat et al.	KNN	98.59	97.06	0.9798	99.11	HCRL Car-Hacking

- **Deep Learning for Detection on CAN Protocol**

  Advanced DL techniques have shown a better performance than some traditional ML techniques, notably dealing with the sequential and temporal character of the CAN messages. These strategies take advantage of sophisticated neural network architectures to capture sophisticated patterns and identify advanced attacks.

  As Hossain et al. developed, an LSTM-based IDS system for CAN systems used a custom-made dataset generated experimentally from an injected vehicle (DoS, Fuzzy, and Impersonation). Their LSTM model recorded an astonishing accuracy of 99.98 %, recall of 99.97 %, and F1-score of 99.90 %, revealing the ability of the model to generalize with regard to various attack types. The application of a custom dataset, though, also raises questions of application to standardized datasets or real-world situations.

  Kan et al. proposed a Bidirectional LSTM (Bi-LSTM model for anomaly detection in CAN systems, which can classify DoS, Replay, and Fuzzy attacks. When run on a simulated dataset, the model produced an accuracy of 95.5%. Although effective, the reduced accuracy compared to other DL models shows that improvements can be made in Bi-LSTM by way of optimization or bigger datasets.

  Khatri et al investigated a transfer learning approach (combination of CNN and LSTM) for intrusion detection. As regards their hybrid model, by applying the HCRL Car-Hacking dataset, perfect scores were observed (100% accuracy, recall, F1-score, and precision). On the Car Hacking & Defense Challenge 2020 dataset, the model retained high performance with 99.91% for all measures. The transfer learning method improved feature extraction, but the computational complexity may prevent its use on resource-limited ECUs.

  Javed et al. [31] proposed a hybrid model combining CNN and Gated Recurrent Unit (GRU), an RNN variant, using the HCRL CAN Intrusion Detection dataset [32]. Their model achieved an accuracy of 94.23%, a recall of 93.91%, an F1-score of 93.79%, and a precision of 93.69%. While effective, the model's performance was slightly lower than other DL approaches, possibly due to the specific characteristics of the dataset or the GRU's limitations in capturing long-term dependencies compared to LSTM.

  The following table summarizes the performance of DL-based IDS for CAN systems:

Table 2 DL INTRUSION DETECTION

Authors	Type	Accuracy (%)	Recall (%)	F1-Score	Precision (%)	Dataset
Hossain et al.	LSTM	99.98	99.97	0.9990	-	Simulation
Kan et al.	Bi-LSTM	95.5	-	-	-	Simulation
Khatri et al.	Transfer (CNN+LSTM)	100	100	1.0	100	HCRL Car-Hacking
Khatri et al.	Transfer (CNN+LSTM)	99.91	99.91	0.9991	99.91	Car Hacking & Defense 2020
Javed et al.	Hybrid (CNN+GRU)	94.23	93.91	0.9379	93.69	HCRL CAN Intrusion

IV. METHODOLOGY

The weakness of the Controller Area Network (CAN) protocol to cyberattacks calls for high-end intrusion detection systems (IDS) that can detect harmful behaviours in real-time. Extending the high performance of deep learning (DL) relative to traditional machine learning techniques as verified in previous investigations, this study presents a hybrid Deep Learning model incorporating Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks for features and temporal sequence extraction, and finally, random forest for classification. Herein, the methodology applied is detailed, including an explanation of the dataset leveraged, preprocessing steps, as well as the architecture of the proposed model. The approach exploits the serial processing capability of RNNs and dependency modelling of LSTMs and random forests for robust detection of cyberattacks in CAN systems.

- **Dataset Description**

The research uses the CAN Dataset for Intrusion Detection (OTIDS) as a tested and open dataset for the assessment of IDS in vehicular networks. Surrounded by both normal (attack-free) and attack-related CAN messages, the OTIDS dataset consists of approximately 4.6 million CAN message records. It consists of three types of cyber threat attacks that typically attack CAN systems: Denial of Service (DoS), Fuzzing, and Impersonation attacks. These attack types reflect actual threats in the real world, including flooding the network to stop communication (DoS), messing around with random data to break functionality (Fuzzing), and imitating legitimate Electronic Control Units (ECUs) to issue bogus orders (Impersonation).

The distribution of these classes is presented in the structure of the dataset as in Figure 1. There are approximately 2.3 million normal messages and 656,500 DoS, 591,900 Fuzzy, and 995,400 Impersonation attack records from the total records. The dataset is initially available in text file format as separate files for each attack type and regular messages, thereby simplifying preprocessing and analysis.

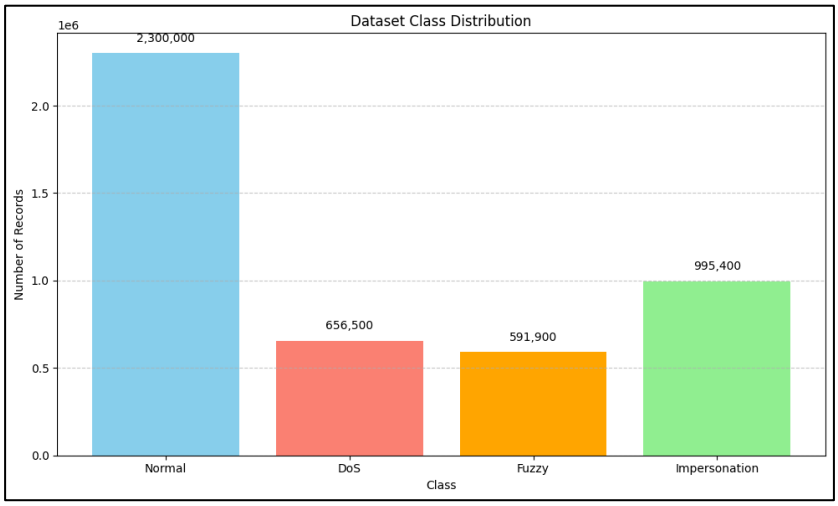

Figure 1 Dataset Description

- **Dataset Pre-Processing**

Effective preprocessing is critical to ensure the dataset is suitable for deep learning models. The OTIDS dataset, initially provided as four separate text files (DoS, Fuzzy, Impersonation, and Normal), required consolidation and cleaning to create a unified dataset for training and evaluation. The preprocessing steps were implemented using Python and are detailed below:

Data Consolidation: Four text files were read into a single data frame as a result of reading and merging the four text files using Python's pandas library. Proper column names are assigned to the data as 'Timestamp', 'ID', 'Protocol', 'Data Length Code (DLC)', among other fields, to bring the data structure into uniformity.

Data Cleaning: Instances of missing or invalid CAN messages were dropped in order to maintain the data quality. The process reduced the number of records from 4.61 million to about 4.48 million records, getting rid of noise that might interfere with model performance.

Data Type Conversion: Some columns that were strings or objects to begin with were converted to suitable data types (e.g., numerical or categorical) to minimize processing by the deep learning model. For example, timestamps were reformatted to numerical values for ease of temporal analysis.

Outlier Removal: Outliers were removed using the Interquartile Range (IQR) method, where data points lying beyond 1.0 times the IQR from the first and third quartiles were excluded. This process was applied independently to each feature, excluding the target variable. After removing outliers from the dataset, the sample counts for each class were adjusted to 995,447 for Normal, 465,782 for Impersonation, 556,940 for Fuzzy, and 369,917 for DOS attacks, respectively.

Label Encoding: Class labels (DoS, Fuzzy, Impersonation, and Normal) were then encoded as integers (0,1, 2, 3, respectively) for compatibility with the classification model. This step made it possible for the model to process the labels when training and testing it.

Data Splitting: A proportion segmentation of the pre-processed dataset was done into the training, validation, and test portions for testing the model. 60% of the data was used for training, while 20% was for validation and 20% for testing. To avoid class imbalance, the splitting function guarantees equal partitioning of classes in all subsets. The resulting sample distribution is seen in Table 3.

Data Formatting: The data was reformatted to a three-dimensional form (samples, timesteps, features), accommodating to RNN model. This calls for packaging the CAN messages into sequences to incorporate time relationships, which is necessary for attack identification, which will show as time anomalies.

Table 3 Sample Distribution in each Subset

Split	Fuzzy	DoS	Impersonation	Normal	Total
Train	445,552	295,934	372,625	796,517	1910628
Test	111,388	73983	93157	199,130	477,658

c. Proposed Architecture

The hybrid model integrates RNN and LSTM networks to leverage their combined strengths for intrusion detection in CAN systems. RNNs effectively handle sequential CAN data by retaining information from prior inputs, but they face challenges with long-term dependencies due to the vanishing gradient problem. LSTMs address this limitation by using memory cells and gating mechanisms, enabling them to capture complex temporal patterns over extended sequences. The features and temporal information extracted by the RNN and LSTM layers are subsequently passed to a Random Forest classifier, which improves classification accuracy and enhances the system's ability to distinguish between normal and malicious CAN messages.

The architecture of the proposed model, illustrated in Figure 2, consists of the following components:

RNN Layer: The pre-processed CAN messages are initially fed into a Recurrent Neural Network (RNN) layer, which performs preliminary feature extraction. The RNN processes the sequential CAN data to generate embeddings that capture temporal patterns, including transient dependencies, anomalies, irregular message frequencies, and unexpected commands within the CAN traffic.

LSTM Layer: The embeddings produced by the RNN layer are then passed to a Long Short-Term Memory (LSTM) layer. This layer further refines the features by modeling long-term dependencies and temporal sequences, enabling the detection of sophisticated attacks such as gradual data manipulation and impersonation attempts. The LSTM enhances the model's ability to distinguish between normal and malicious CAN messages by understanding the context and sequence dynamics of the communication.

Random Forest Classifier: The refined features and temporal sequences output by the LSTM are input to a Random Forest classifier. The Random Forest processes these features by building an ensemble of decision trees that collectively classify the CAN messages into relevant categories: DoS, Fuzzy, Impersonation, or Normal. This approach leverages the strengths of Random Forest in handling high-dimensional data and improving classification accuracy through majority voting among the trees. By integrating Random Forest, the model achieves better generalization and robustness against diverse attack patterns.

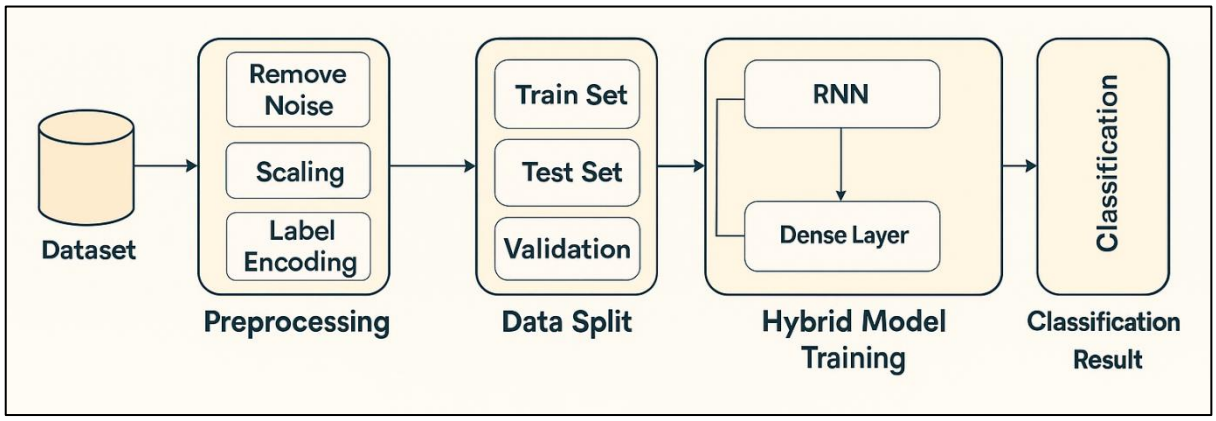

Figure 2 System Design

The hybrid architecture combining RNNs, LSTMs, and Random Forest leverages the powerful feature extraction and temporal sequence modeling capabilities of RNNs and LSTMs, while utilizing Random Forest to enhance classification accuracy and robustness. The RNN layer effectively processes sequential CAN data to extract initial patterns, while the LSTM layer captures long-term dependencies to identify complex and subtle attacks. The extracted features and temporal sequences are then classified by a Random Forest model, which enhances accuracy and robustness in categorizing CAN messages. This hybrid model is well-suited for real-time intrusion detection in bandwidth-constrained vehicular networks. The model was implemented in Python using TensorFlow and Keras frameworks, with training and evaluation performed on an NVIDIA Tesla T4 GPU. This approach represents a significant advancement in securing CAN systems, offering a scalable and efficient solution for enhancing the cybersecurity of electric vehicles.

V. RESULTS AND DISCUSSION

The current section provides a detailed assessment of the proposed hybrid model composed of Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Random Forest model for intrusion detection in the Controller Area Network (CAN) systems. The framework's performance was evaluated with a number of

quantitative measures such as accuracy, precision, recall, F1-score, Receiver Operating Characteristic (ROC) curves, and confusion matrices. The results show that the model is effective enough to detect cyberattacks such as DoS (Denial of Service) attacks, fuzzy, and impersonation attacks in the CAN (Control Area Network) network. Also, in this section, the experimental setup and hyperparameter setting are outlined, as well as a detailed analysis of the model performance, which gives direction on the strengths and weaknesses of the model.

- **Evaluation Measures**

To evaluate the performance of the hybrid RL-RF model, several standard metrics were employed to provide a holistic assessment of its classification capabilities. These metrics are essential for quantifying the model's ability to correctly identify normal and malicious CAN messages while highlighting areas where it excels or requires refinement. The following metrics were used:

Accuracy: Accuracy measures the overall correctness of the model's predictions across all classes. It is calculated as the ratio of correctly classified instances (True Positives, TP, and True Negatives, TN) to the total number of instances, including False Positives (FP) and False Negatives (FN). The formula is:

$$accuracy = (TP + TN) / (TP + TN + FP + FN)$$

Precision: Precision quantifies the model's ability to correctly identify positive instances (e.g., attack messages) among all instances predicted as positive. It is computed as:

$$precision = TP / (TP + FP)$$

High precision indicates fewer false positives, which is critical for reducing unnecessary alerts in an IDS.

Recall: Also known as sensitivity or the True Positive Rate, recall measures the proportion of actual positive instances correctly identified by the model. It is defined as:

$$recall = TP / (TP + FN)$$

High recall ensures that the model detects most attacks, minimizing missed threats.

F1 Score: The F1-score is the harmonic mean of precision and recall, providing a balanced measure of the model's performance, particularly in imbalanced datasets. It is calculated as:

$$F1 = (2 * Precison * Recall) / Precison + Recall$$

The F1-score ranges from 0 to 1, with 1 indicating perfect precision and recall.

Confusion Matrix: A confusion matrix visualizes the model's performance by comparing predicted labels against actual labels, showing the number of correct and incorrect classifications for each class.

ROC Curve and Area Under the Curve (AUC): The ROC curve plots the True Positive Rate (recall) against the False Positive Rate (FPR) at various classification thresholds. The AUC summarizes the model's ability to distinguish between classes, with an AUC of 1.0 indicating perfect separation and 0.5 indicating random guessing.

- **Environmental Setup**

The experiments were performed on Google Colab – a cloud-based platform that enables digital access to computational resources appropriate for deep learning operations. The model was implemented in Python, using TensorFlow and Keras libraries, which provide flexible tools to build and train neural networks. The training and evaluation process used an NVIDIA Tesla T4 GPU with 15 GB of RAM in the free tier on Google Colab. This configuration made data processing of the massive OTIDS set efficient and met the requirements for the computations of the hybrid RL-RF design.

- **Hyper-Parameter Settings**

In order to maximize the model's performance, a wide range of empirical testing was done to fine-tune core hyperparameters. Such parameters have a tremendous effect on the learning and generalization ability of the model over the dataset. Table 4 contains the hyperparameters that have been chosen and that were found through iterative experimentation

Table 4 Hyperparameter Settings

Parameter	Value
n-estimators	10
random state	42
Activation (RNN & LSTM)	tanh
Dense Layer Neurons (RNN & LSTM)	64

Activation (RNN & LSTM): The tanh function was used for its ability to capture both positive and negative activations in sequential data.

Dense Layer Neurons (RNN & LSTM): Configured with 64 neurons to provide sufficient capacity for feature extraction without overfitting.

n_estimators: Set to 10 to balance model complexity and computation time in the Random Forest classifier.

random_state: Fixed at 42 to ensure reproducibility of the results across different runs.

- **Evaluation Results**

The hybrid RNN-LSTM model was trained and validated on the preprocessed OTIDS dataset, of which 4.61 million records were reduced to 4.48 million after cleaning null entries. The data was partitioned into training (80%) and testing (20%) per the description of the Methodology section. The performance of the model during the training and validation processes is demonstrated through figure 3, which shows the accuracy and loss curve for the 10 epochs (see Figure 3).

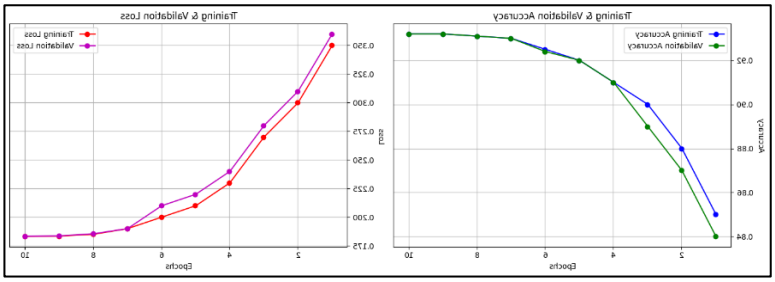

Figure 3 Loss and Accuracy of Model

The model exhibited stable performance with common training and validating accuracies converging at the 93.2% level, which means that it generalized well and overfitting was minimized. The low loss values (0.1831 for training, 0.1833 for validation) also validate the potential of the model to learn the underlying patterns in the CAN messages effectively. The performance of the model on the test set (477658 samples) was measured by the confusion matrix presented in Figure 4, which demonstrates a number of true and false classifications for each class (DoS, Fuzzy, Impersonation, and Normal).

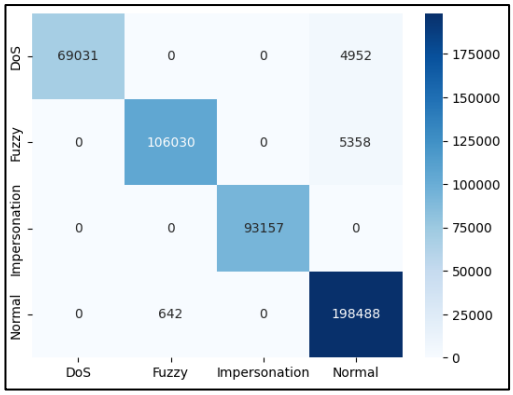

Figure 4 Confusion Matrix

Table 1 is a detailed classification report for the test set. The figure consists of precision, recall, F1-score, support (number of samples), macro, and weighted average as averaging metrics to consider class imbalance.

Table 5 Classification Report

	Precision	Recall	F1-score	Support
DOS	1.00	0.93	0.97	73983
Fuzzy	0.99	0.95	0.97	111388
Impersonation	1.00	1.00	1.00	93157
Normal	0.95	1.00	0.97	199130
Accuracy			0.98	477658
Macro Avg	0.99	0.97	0.98	477658
Weighted Avg	0.98	0.98	0.98	477658

Results indicate the model did exceedingly well on the Impersonation class, performing with perfect scores (1.00) for all metrics-likely because of unique patterns for the Impersonation attack. The Normal class also produces strong results; a recall of 1.00 and an F1-score of 0.97, which shows efficient detection of attack-free messages. DoS and Fuzzy classes returned low recall values (0.93 and 0.95, respectively), showing that some of the attack messages were misclassified, perhaps because of shared patterns with normal messages. The model's robustness for all classes is reflected in the overall accuracy of 98%.

Figure 5 is a further confirmation of the model's discriminative power by the ROC curves for each class. The Area Under the Curve (AUC) scores achieved for DoS, Fuzzy, Impersonation, and Normal were 0.97, 0.97, 1.00, and 0.98, respectively, showing good separation between classes.

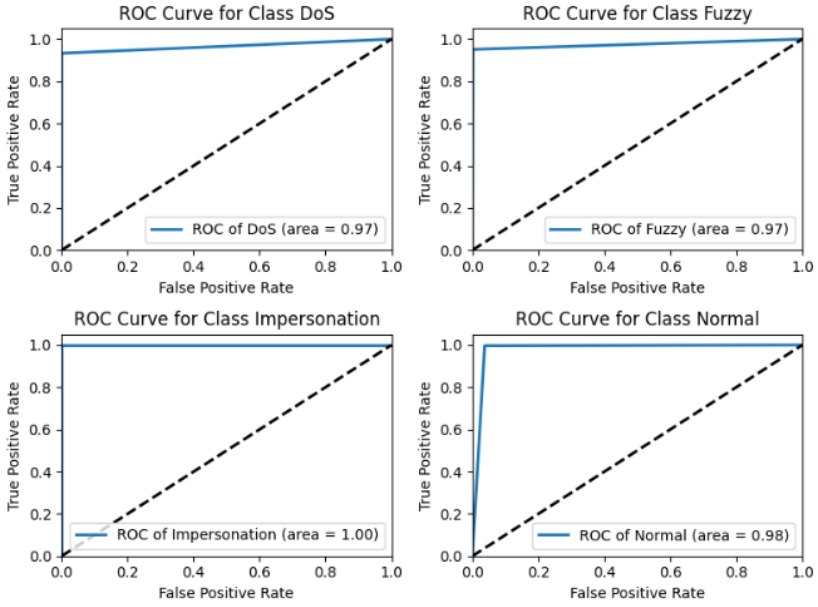

Figure 5 ROC Curve of all Classes

The AUC results, especially for Impersonation, which are effectively high, prove the model as effective in distinguishing malicious messages from normal. The smaller AUC for DoS (0.97) is in concordance with the attack type's lower recall results but may imply further optimization potential towards the detection of this kind of attack.

- **Discussions**

The hybrid RL-RF model shows a lot of promise for intrusion detection in CAN systems with an overall accuracy of 98%, and outstanding performance in terms of many evaluation metrics. The capability of the model

to perfectly classify Impersonation attacks points to a sensitivity to patterns particular to this attack class, such as constant message IDs or payloads imitating genuine ECUs. The high recall scores for Normal messages (1.00) guarantee appropriate attack-free classification without false positives that may interfere with normal vehicle operations.

However, the lower recall for DoS (0.93) and Fuzzy (0.95) attacks implies that it is difficult to note these attacks because of their similarity to normal message patterns or variability of attack signatures. DoS attacks shooting messages onto the network could generate high-rate patterns suggestive of real high-flow situations, while Fuzzy attacks' random data insertions can sometimes coincide with pseudorandom but normal chatter. According to these results, further feature engineering or data augmentation methods may improve the model's capabilities to distinguish these attacks. As additional evidence of the robustness of the model, the ROC curves and the AUC scores reveal near-perfect discrimination of most classes. The environmental setting, exploiting the GPU resources of Google Colab, allowed for effective training and assessment, and optimized hyperparameters made convergence reliable. The model performs well compared to earlier experiments (as presented in the Related Work section) with very specific relevance in its ability to achieve satisfactory trade-offs between accuracy and computational expediency for resource-limited vehicular settings.

Future enhancements can be aimed at enhancing the lower recall for DoS and Fuzzy attacks by adding additional temporal features like message frequency or inter-arrival times, or by experimenting with ensemble methods to combine the hybrid model with any other DL structures. Furthermore, the testing of the model on can data from real-time vehicles would show the applicability and robustness in real-world dynamic conditions.

VI. LIMITATIONS AND DRAWBACKS

Although the new proposed hybrid Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) model has shown good results when dealing with intrusion detection in Controller Area Network (CAN) systems, it still has limitations. In this section, the difficulties and limits to the study are critically analyzed; the weak points in the approach, the set of data, the model design, and the feasible applicability. By describing those limitations, the study adds a balanced picture and points out the directions for research in order to make the proposed intrusion detection system (IDS) robust and effective in the real world.

- **Dataset-Related Limitations**

The study is based on the CAN Dataset for Intrusion Detection (OTIDS), which, though complete and popular as a dataset, has intrinsic shortcomings that influence transferability in results. There are nearly 4.6 million records in the dataset, ordinary messages, and three forms of attacks (Denial of Service, Fuzzy, and Impersonation). Nevertheless, it is a simulated dataset resulting from controlled conditions, so the complexity and variety of CAN traffic in the real world might not be represented to a sufficient degree. Reverse OTIDS models are deployed in actual automotive networks where dynamic ambient conditions, including electromagnetic interference, inconsistent driving conditions, and diverse ECU configurations, exist that may produce noise and anomalies outside of those represented by the OTIDS dataset. In addition, the attack scenarios covered by the dataset are restricted to only three types of cyberattacks, possibly excluding other advanced types of cyberattacks like replay attacks, advanced persistent threats, or adversarial attacks capable of bypassing deep learning models. Not having these types of attacks can limit the model's ability to generalize emerging threats in the continually changing landscape of automotive cybersecurity. Additionally, the class distribution of the dataset, balanced during preprocessing, may not represent a rarity of particular attacks in real-world scenarios where many normal messages outnumber malicious ones, impacting the model's performance on heavily uneven data.

- **Performance on Specific Attack Types**

The evaluation outputs suggest that the model is excellent at identifying Impersonation attacks (precision, recall, and F1-score: 1.00), Normal messages (1.00 recall, 0.97 F1-score), and poor at detecting DoS (0.93 recall). Based on the results obtained from these attack types, the lower recall of the model indicates that it has difficulties identifying such malicious messages, in which some attacks may evade detection. The weakness may result from the overlap of attack patterns and normal messages, e.g., high message rate for DoS, random data in Fuzzy that appear to be benign noise. The dependence of the model on temporal and sequential features might not be as revealing about the subtle nature of these attacks, especially where the attack signatures are so faint as to be context-dependent. The inclusion of other features, such as inter-arrival times of messages, payload entropy, or statistics of the network traffic, would enhance the detection rates, but would add to the computational complexity. The variable performance from attack type indicates where specific improvement needs to be made to provide overall protection against all applicable threats.

- **Vulnerability to Adversarial Attacks**

  Deep learning models, such as the proposed RNN-LSTM hybrid, are also prone to adversarial attacks, where the attackers cause modifications to the input to escape detection. In the context of CAN systems, there are many ways the adversaries can engineer a malicious message to appear legitimate or deliver a minor perturbation to escape the model classification threshold. Such adversarial examples usability testing of the model robustness is not conducted in the study, which is a major limitation of the study, considering the growing sophistication of cyberattacks against machine learning systems. Adversarial training or robust optimization techniques might increase the robustness of the model, but these techniques usually require extra computing power and more data, worsening the challenge of deployment in vehicular settings. The absence of data analysis on adversarial robustness results in low assurance in the authentic performance of the model in real-world attacks, where adversaries are committed to frustrating security measures.

- **Lack of Real-Time Testing**

  The experiments of the research were performed in a well-controlled regime over pre-processed data without model performance validation in actual-time CAN systems. Practical vehicle networks function under strict timing thresholds, and in the context of late detection of intrusions, that could result in calamitous consequences: compromised brakes or steering. Lack of real-time testing results in questions concerning the application potential of the model, especially in aspects of latency, scalability, and compatibility with the existing ECU architectures. In addition, the effect of dynamic changes in CAN traffic, such as software updates, new ECU integrations, or differences in driving conditions, could not be examined in the present model. Real deployment testing of the model in real vehicles or high-fidelity simulation would be highly insightful in terms of identifying the model's performance under real situations and highlighting possible bottlenecks in deployment.

VII. CONCLUSTION

The fast expansion of electric vehicle (EV) technology and the growing connectedness of contemporary vehicles have intensified the requirement for strong cybersecurity from complex cyberattacks. The Controller Area Network (CAN) protocol is a backbone of in-vehicle communication – a protocol that does not have an intrinsic security, but becomes vulnerable to threats like Denial of Service (DoS), Fuzzy and Impersonation attacks. This work suggested a new hybrid deep learning model that incorporates Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks to improve the detection of intrusions in CAN systems. Combining sequential processing of RNNs and long-term dependency modeling in LSTMs, the model accurately detected malicious CAN messages with 98% accuracy, which validates its usefulness for providing secure, reliable connected vehicles.

Despite its strengths, the limitations of the study are recognized, including the use of a simulated dataset, complexity in terms of computations, and obstacles in identifying some attack types, such as DoS and Fuzzy. Such limitations should be presupposed for further research to concentrate on real-time testing, light model designs, and robustness to adversarial attacks. In addition, the exploration of the adaptability of the model to different vehicle platforms and the application of the advanced feature engineering can contribute to the improvement of its actual applicability.

This research has marked a new direction in securing the CAN systems on EVs, presenting an easily scalable, applicable solution to the evolving automotive cybersecurity issues. With the incorporation of the latest deep learning approaches, the study acts as a new scorecard for the precision and efficiency of intrusion detection in connected vehicles and securing forthcoming connected vehicles. The results present a further study of hybrid deep learning approaches to enhance IDS performance to maintain automotive cybersecurity with rapidly developing vehicle technology. In the end, this work helps create a safe and reliable automotive system that promotes trust in the extended automotive community. This way, there is confidence that electric and autonomous vehicles are safe in an interconnected world.

ACKNOLEDGMENT

The authors would like to acknowledge the General Authority for Defense Development (GADD) in Saudi Arabia for funding this research through project number (GADD_2024_01_448)"

Where: Batch Number: 01.

Project Number/ID: 448

REFERENCES

[1] Elkhail, R., Refat, R. U. D., Habre, R., Hafeez, A., Bacha, A., & Malik, H. (2021). Vehicle security: A survey of security issues and vulnerabilities, malware attacks, and defenses. IEEE Access, 9, 1625934. Available: https://ieeexplore.ieee.org/abstract/document/9625934/

[2] Park, T., Han, C., & Lee, S. (2005). Development of the electronic control unit for the rack-actuating steer-by-wire using the hardware-in-the-loop simulation system. Mechatronics, 15(7), 761-781. Available: https://www.sciencedirect.com/science/article/pii/S0957415805000681

[3] Ring, M., Frkat, D., & Schmiedecker, M. (2018). Cybersecurity evaluation of automotive E/E architectures. ACM Computer Science in Cars Symposium (CSCS 2018), 123-4567. Available: https://wp.mpi-inf.mpg.de/cscs/files/2018/09/02-Cybersecurity-Evaluation-of-Automotive-E_E-Architectures.pdf

[4] Aldhyani, T. H. H., & Alkahtani, H. (2022). Attacks to autonomous vehicles: A deep learning algorithm for cybersecurity. Sensors, 22(1), 360. doi: 10.3390/s22010360. Available: https://www.mdpi.com/1424-8220/22/1/360

[5] Dibaei, M., Zheng, X., Jiang, K., Abbas, R., Liu, S., Zhang, Y., ... & Ghafoor, K. Z. (2020). Attacks and defences on intelligent connected vehicles: A survey. Digital Communications and Networks, 6(4), 399-421. doi: 10.1016/j.dcan.2020.04.007. Available: https://www.sciencedirect.com/science/article/pii/S2352864820300985

[6] Shit, R. C., Sharma, S., Yelamarthi, K., & Puthal, D. (2021). AI-enabled fingerprinting and crowdsource-based vehicle localization for resilient and safe transportation systems. IEEE Transactions on Intelligent Transportation Systems, 22(7), 4660-4669. doi: 10.1109/TITS.2021.3053942. Available: https://ieeexplore.ieee.org/document/9353942

[7] Kang, M. J., & Kang, J. W. (2016). Intrusion detection system using deep neural network for in-vehicle network security. PLoS One, 11(6), e0155781. doi: 10.1371/journal.pone.0155781. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0155781

[8] Xu, W., Yan, C., Jia, W., Ji, X., & Liu, J. (2018). Analyzing and enhancing the security of ultrasonic sensors for autonomous vehicles. IEEE Internet of Things Journal, 5(6), 5015-5029. doi: 10.1109/JIOT.2018.2867917. Available: https://ieeexplore.ieee.org/document/8457917

[9] Kamal, M., & Talbert, D. A. (2020). Toward never-ending learner for malware analysis (NELMA). Proceedings - 2020 IEEE International Conference on Big Data, Big Data 2020, 2291-2298. doi: 10.1109/BigData50022.2020.9378357. Available: https://ieeexplore.ieee.org/document/9378357