HybridSecNet: In-Vehicle Security on Controller Area Networks Through a Hybrid Two-Step LSTM-CNN Model

Amit Chougule, Ishan Kulkarni, Tejasvi Alladi, *Senior Member, IEEE*, Vinay Chamola, *Senior Member, IEEE*, and Fei Richard Yu, *Fellow, IEEE*

***Abstract*—The modern Intelligent Vehicle (IV) is a complex technological marvel that heavily relies on the Controller Area Network (CAN) bus system to enable seamless communication among different electronic control units (ECUs). However, the CAN bus system lacks security mechanisms for authentication and authorization, leaving it vulnerable to various attacks. Malicious actors can freely broadcast CAN messages without protection, making the system susceptible to DoS, Fuzzing, and Spoofing attacks. Therefore, it is crucial to devise methods to safeguard modern vehicles from such threats. In this research paper, we introduce HybridSecNet, A hybrid two-step LSTM-CNN Model for Intrusion Detection, a deep learning-based architecture specifically designed to bolster in-vehicle security on Controller Area Networks (CAN). HybridSecNet comprises two stages of classification: the first stage employs long short-term memory (LSTM) to categorize input data as either normal or attacked, and the second stage further classifies the attacks into specific types using Convolutional Neural Networks (CNN). This two-step approach significantly enhances classification accuracy and reliability, yielding remarkable results with accuracy, precision, recall, and an F1-score of approximately 99.5% for CAN bus network attacks. Comparative analyses with existing single-step models underscore the superiority of our proposed model, demonstrating its potential to revolutionize in-vehicle security in the realm of modern intelligent vehicles.**

***Index Terms*—Controller area network, intrusion detection system, intelligent transport system.**

I. Introduction

INTELLIGENT Vehicle Network (IVN) plays a crucial role in the automotive industry by enabling efficient and reliable data exchange among various electronic components within vehicles. The use of these protocols has become increasingly important as modern cars are equipped with a wide array of electronic systems, including engine control units, infotainment systems, safety systems, and more [1], [2], [3]. These protocols facilitate real-time communication, allowing different modules to work together seamlessly, enhancing overall vehicle performance, safety, and user experience. In the context of autonomous driving cars, the significance of serial communication protocols becomes even more pronounced due to the complexity of autonomous vehicles. Seamless and instantaneous data transmission between sensors, actuators, and control units is essential to ensure precise decision-making and execution in autonomous driving. With advanced driver assistance systems and autonomous functionalities, vehicles must process vast amounts of sensor data in real time to navigate, perceive the environment, and react swiftly to changing road conditions.

Amidst various serial communication protocols used in the automotive industry, Controller Area Network (CAN) stands out as the most significant and useful one for autonomous driving cars and traditional cars due to several key advantages [4], [5], [6]. Firstly, CAN is renowned for its affordability, making it a cost-effective option for vehicle manufacturers. Its widespread adoption in the automotive domain has resulted in economies of scale, further driving down costs. Another notable advantage of CAN is its exceptional reliability, specifically designed to operate reliably in harsh automotive environments [4], [7], [8]. It is capable of maintaining consistent communication even in the presence of electrical noise and interference, ensuring robust and uninterrupted data transmission [4], [9], [10]. Scalability is yet another strength of CAN. It is suitable for a broad spectrum of applications, ranging from simple interior functions to complex safety-critical systems. This versatility allows CAN to be integrated into various components and subsystems within a vehicle, facilitating efficient communication and coordination between them. Furthermore, CAN benefits from international standardization, which ensures compatibility and seamless communication between devices and systems that utilize the CAN protocol. This standardized approach fosters interoperability, enabling different vehicles and components from various manufacturers to communicate effectively and share information. These collective advantages position CAN as the preferred

Manuscript received 24 August 2023; revised 31 January 2024 and 9 April 2024; accepted 3 June 2024. Date of publication 13 June 2024; date of current version 17 October 2024. The work of Vinay Chamola and Fei Richard Yu was supported by SICI SICRG. The review of this article was coordinated by Dr. Ying He. *(Corresponding author: Vinay Chamola.)*

Amit Chougule and Ishan Kulkarni are with the Department of Electrical & Electronics Engineering, Birla Institute of Technology And Science - Pilani, Pilani 333031, India (e-mail: amitchougule121@gmail.com; f20201802@pilani.bits-pilani.ac.in).

Tejasvi Alladi is with the Department of Computer Science and Information Systems, Birla Institute of Technology And Science - Pilani, Pilani 333031, India (e-mail: tejasvi.alladi@pilani.bits-pilani.ac.in).

Vinay Chamola is with the Department of Electrical and Electronics Engineering & APPCAIR, Birla Institute of Technology And Science - Pilani, Pilani 333031, India (e-mail: vinay.chamola@pilani.bits-pilani.ac.in).

Fei Richard Yu is with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada (e-mail: richard.yu@carleton.ca).

Digital Object Identifier 10.1109/TVT.2024.3413849

communication protocol for a wide range of automotive applications.

In addition to the widely used Controller Area Network (CAN), the automotive industry employs various other communication protocols in vehicles to cater to different requirements and applications. Among these protocols are LIN (Local Interconnect Network) and FlexRay [11], [12], [13], each with its distinct characteristics and functionalities. LIN is a low-cost technology commonly employed for basic networking within vehicles. With a bandwidth of up to 20 Kbps, it finds primary usage in body and interior applications. However, its limited capability to process complex data renders it less suitable for advanced functionalities requiring higher processing capacity. On the other hand, FlexRay, a faster communication protocol compared to CAN, can handle more sophisticated data and is frequently utilized in advanced driver assistance systems (ADAS) and safety-critical applications. Boasting a high bandwidth of up to 10 Mbps, FlexRay facilitates the transmission of large amounts of data. Nevertheless, due to its increased complexity and cost, FlexRay has not achieved the same level of widespread adoption as CAN. Ethernet, traditionally used in computer networks, is also gaining traction as a communication protocol within the automotive domain [14], [15], [16]. It enables information exchange between various nodes such as vehicles, charging stations, or base stations [14], [17], [18]. With a bandwidth of up to 1 Gbps, Ethernet exhibits significant potential for handling massive amounts of data. However, its reliability and determinism fall short when compared to CAN, thereby rendering it unsuitable for safety-critical functions.

While CAN (Controller Area Network) offers numerous advantages, it is not without limitations that render it vulnerable to autonomous attacks. One key limitation is the absence of encryption for CAN messages [19], [20], [21]. The lack of encryption means that anyone with access to the CAN bus can intercept and manipulate the transmitted data, opening the door for attacks such as spoofing or replay attacks. This vulnerability poses a significant risk to the integrity and security of the communication system. Moreover, CAN has limited authentication capabilities [22], [23], [24]. Messages transmitted over the CAN bus lack robust authentication mechanisms, making it challenging to verify the identity of the sender or receiver. This limitation provides an opportunity for attackers to impersonate legitimate Electronic Control Units (ECUs) and carry out malicious actions without detection. As a result, attackers can propagate their attacks across the entire network, amplifying their potential impact and increasing the difficulty of identifying and mitigating their activities. Furthermore, CAN's error detection capabilities are also limited [25], [26], [27]. This means that errors or faults in the network may go unnoticed, making it more challenging to identify and address attacks that exploit these vulnerabilities. The lack of comprehensive error detection mechanisms increases the risk of undetected attacks and compromises the overall security of the CAN-based communication system. To ensure the resilience and security of autonomous systems, it is crucial to address these limitations and implement additional security measures. Encryption, authentication protocols, and robust error detection mechanisms should be considered to mitigate the risks associated with autonomous attacks on CAN-based networks. By addressing these vulnerabilities, the integrity, confidentiality, and authenticity of the communication system can be strengthened, enhancing the overall security posture of autonomous vehicles and other CAN-enabled systems.

Given the limitations of CAN and the growing need for robust security measures, numerous Intrusion Detection techniques have been proposed in the literature, as discussed in Section II. Among these techniques, Machine Learning has shown great promise in securing Intelligent Vehicle Networks (IVNs) [28], [29], [30]. However, while Machine Learning-based Intrusion Detection Systems (IDS) exhibit high accuracy in detecting attacks, they often struggle with accurately classifying the specific type of attack. To address this challenge, we propose a two-step IDS based on machine learning that offers enhanced classification capabilities with reliability. Our approach goes beyond the classification of data as either normal or malicious and aims to determine the nature or specific type of attack with greater precision. By incorporating an additional step into the detection process, we were able to achieve significantly improved accuracy with reliability in identifying and classifying different types of attacks. This study presents a comprehensive and robust Intrusion Detection model specifically designed for detecting intrusions in Intelligent Vehicle Networks (IVNs). The proposed model offers several significant contributions to the field, including:

1) Our research proposes a novel Two-step Intrusion Detection System (IDS) designed explicitly for Intelligent Vehicle Networks (IVNs).
2) In comparison to existing techniques, our proposed technique offers significant advantages, particularly in terms of higher classification accuracy and improved reliability.
3) HybridSecNet exhibits resource-saving and real-time capabilities due to its efficient design. When the input data is not identified as an attack type in the first step, the second step which is responsible for determining the attack type is not invoked. This feature ensures that the second step is utilized only when necessary. As a result, unnecessary computational resources are conserved, leading to faster and more efficient processing of data.
4) We evaluate our proposed Two-step IDS by comparing it with a Single-step IDS model. Using metrics like Accuracy, Precision, Recall, and F1 score, we assess the effectiveness of both models in detecting and classifying attacks within IVNs. Our comparative analysis shows that the Two-step model outperforms the Single-step model in terms of classification accuracy and robustness.

In this paper, we have comprehensively explored various sections, as outlined below. Section I serves as an introduction, providing a comprehensive overview of the research topic. In Section II, we delve into existing literature and related works, examining the solutions and methodologies employed in the field. Section III presents a preliminary background of this study, laying the foundation for further investigation. In Section IV, we propose our innovative two-step intrusion detection architecture, detailing its design and components. A thorough discussion of the simulation environment is presented in Section V. The

results and analysis of our work are presented in Section VI, where we analyze the performance of our proposed model and its efficacy in detecting intrusions. Finally, Section VII provides a comprehensive conclusion of our research, summarizing the essential findings and contributions and outlining potential future directions for further research in this area.

II. Related Works

In the field of Intrusion Detection System (IDS) countermeasures for in-vehicle systems, various approaches have been proposed. These countermeasures can be broadly categorized into four distinct categories. One category is the Fingerprint Based approach, which focuses on utilizing unique characteristics or signatures to detect and identify attacks in in-vehicle systems. Yang et al. [31] proposed signature-based and anomaly-based IDS techniques. The paper provides insights into the design and implementation of the MTH-IDS system and evaluates its performance through testbed tests. Another notable approach is the clock skew-based IDS discussed in [32]. This real-time vehicle intrusion detection system leverages the analysis of clock skew in electronic control units (ECUs) present in a vehicle. By monitoring and analyzing the clock skew of multiple ECUs, the proposed system can detect various forms of attacks. The work presents the design and implementation of the clock skew-based IDS and highlights its capabilities in detecting and mitigating security threats in in-vehicle systems.

The Parameter Monitoring approach is another category of IDS countermeasures for in-vehicle systems. In this approach, the focus is on monitoring and analyzing various parameters or characteristics of the network traffic to detect anomalies and potential intrusions. An example of the Parameter Monitoring approach is presented in [33]. The paper introduces an intrusion detection system (IDS) called OTIDS, which specifically targets the detection of network traffic irregularities using remote frames. By monitoring and analyzing remote frames, the system aims to identify abnormal network behaviours that may indicate potential intrusions or security threats. In a similar vein, [34] proposes a novel method for detecting in-vehicle network intrusions through the analysis of time intervals between Controller Area Network (CAN) signals. The authors suggest employing the time intervals between CAN signals as a means to detect anomalies in network traffic, which can be indicative of malicious activities. The study presents an overview of the proposed intrusion detection system (IDS) and its components, including the CAN message interval analysis technique and the system architecture.

The Information Theory-based approach is the third category of IDS countermeasures for in-vehicle systems. This approach leverages concepts and techniques from information theory to analyze the characteristics of network traffic and detect potential intrusions. In the study presented in [19], the authors propose a unique sliding window similarity analysis method for in-vehicle network intrusion detection. The objective of this approach is to detect both regular and malicious traffic patterns in real-time, with a specific focus on denial-of-service (DoS) attacks. By analyzing the similarity between sliding windows of network traffic, the method aims to identify deviations from normal patterns that may indicate the presence of DoS attacks or other malicious activities. The paper provides a detailed description of the proposed approach and its components, highlighting its effectiveness in detecting and mitigating DoS attacks in the in-vehicle network environment. Similarly, in [35], the authors present an approach for analyzing the differences between normal and malicious network traffic patterns using the Hamming distance measure. The paper introduces an intrusion detection system (IDS) that employs the Hamming distance to quantify the dissimilarity between observed network traffic and normal traffic patterns. By measuring the Hamming distance, the IDS can identify unusual or malicious traffic patterns that deviate significantly from the expected behavior. The study offers an overview of the proposed IDS and its components, including the Hamming distance method and the system architecture.

The Machine Learning-based approach is the fourth and most popular category of IDS countermeasures for in-vehicle systems. This approach leverages the power of machine learning algorithms and techniques to detect and identify intrusions in the in-vehicle network. In [36], a deep neural network-based anomaly detection approach is proposed. This method collects feature vectors from in-vehicle network data and utilizes a pre-trained deep belief network to train the parameters of the deep neural network. By leveraging the power of deep learning, this approach aims to identify anomalous patterns in the network traffic, indicating potential intrusions or attacks. Another machine learning-based IDS technique is CANtransfer, proposed by [37]. This technique utilizes transfer learning, where knowledge gained from one set of attacks is transferred to improve the detection performance for novel attacks. While it demonstrates enhanced detection performance for novel attacks, it may exhibit poorer detection performance for well-known threats. The study presented by [38] introduces a Generative Adversarial Networks (GANs) based approach called GIDS. This approach monitors differences from previously learned typical behavior and identifies threats by stimulating the system. By utilizing GANs, GIDS achieves a high accuracy of at least 96% in detecting cyber-attacks in Controller Area Networks (CANs). Agrawal, Kushagra, et al. [39] suggested a novel intrusion detection model based on a thresholding and error reconstruction approach using deep learning is proposed. This approach combines deep learning techniques with thresholding and error reconstruction. The results demonstrate high attack detection rates for individual attacks.

Amato, Flora, et al. [40] leverage Neural Networks, specifically MultiLayer Perceptrons (MLPs), in their methodology. The validation is conducted on a real-world dataset exposed to dos, fuzzy, gear, and rpm attacks. Notable findings unveil that, among MLP models tested with 1 to 5 hidden layers, optimal results manifest in models with 1 to 3 hidden layers. In a parallel endeavor, a lightweight attack detection mechanism is proposed in [41] based on CanNet. It is adept at effectively identifying both periodic and aperiodic Denial of Service (DoS) attacks. This lightweight CanNet image classification network is tailor-made for detecting abnormalities in generated CAN images. The mechanism incorporates a heatmap function to

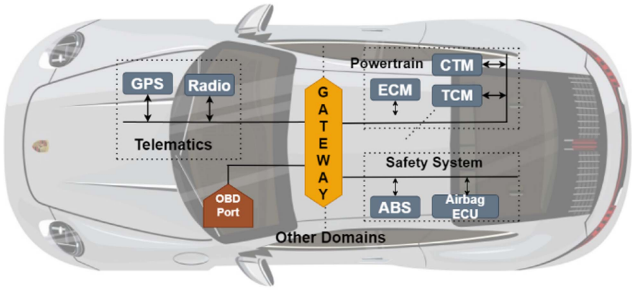


Fig. 1. In vehicle network scenario.

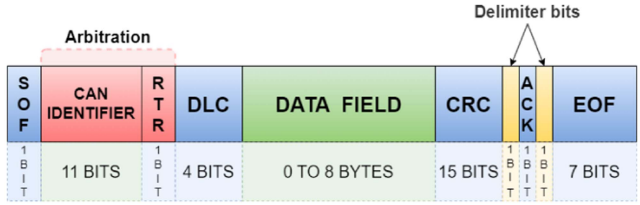


Fig. 2. CAN format.

convert CAN ID values into a 3-bit RGB representation by normalizing the CAN ID. Concurrently, CANShield [42] introduces a deep learning-driven intrusion detection framework for CAN bus signals. Employing multiple Convolutional Neural Network (CNN)-based Autoencoder (AE) models, it operates on various perspectives of the data stream across distinct temporal scales. The process includes a three-step structural analysis of reconstruction losses, culminating in an ensemble to derive the final anomaly score for the CAN bus. Comprising three modules-data preprocessing, data analysis with multiple AE networks, and an attack detection module utilizing an ensemble approach-CANShield ensures comprehensive signal-level monitoring and intrusion detection. In another investigative trajectory, Jeong, Woojin, et al. [43] deploy a Convolutional Neural Network (CNN)-based message source identifier. The performance evaluation involves a channel model designed to replicate real CAN channel characteristics, encompassing both random and authentic CAN data patterns. Furthermore, Javed, Abdul Rehman, et al. [44] propose a technique for in-vehicle intrusion attack on a controller area network, detecting using a CNN and attention-based GRU combination. Similarly, Zhang, Haichun, et al. [45] develop a security evaluation tool for the controller area network.

III. Preliminary Background

An electronic control unit (ECU) is a compact device embedded within a vehicle that is responsible for managing specific functions, ranging from essential tasks like engine and power steering control to comfort features like power windows and music systems [46]. In modern vehicles, multiple ECUs are interconnected through a Controller Area Network (CAN) bus, with each node directly connected to every other node in the network. This decentralized architecture eliminates the need for a central controller, and all nodes on the CAN bus have equal priority [47], [48], [49].

Various systems and components in a car are linked together via an in-vehicle network. This network is depicted in the Fig. 1. GPS and radio communication modules are available in the Telematics area. The Powertrain sector includes modules that govern and control the powertrain components, such as the Engine Control Module (ECM), Transmission Control Module (TCM), and Chassis Control Module (CTM). To ensure vehicle safety, the Safety System includes modules such as the Anti-lock Braking System (ABS) and the Airbag system. This is just an illustration; modern vehicles have many more sections and ECUs. Additionally, the OBD (On-Board Diagnostics) Port allows external devices to obtain diagnostic data. All of these components are linked via a Controller Area Network (CAN) bus, which is controlled by a Gateway. The Gateway acts as a central communication hub, facilitating data flow between many systems. Furthermore, the OBD Port and other domains (such as LIN, Ethernet, and Flexray) are directly connected to the Gateway, allowing for easy integration and communication between various components.

A CAN frame is the fundamental unit of communication in the CAN bus. It consists of a sequence of dominant and recessive bits. The structure of a CAN data frame begins with a start-of-frame (SOF) bit, followed by an 11-bit CAN identifier and a remote transmission request (RTR) bit. The data length code (DLC) field, four bits in length, indicates the number of data bytes present in the data field, which can vary from 0 to 8 bytes. The actual payload of the message is stored within the data field. To detect bit corruption during transmission, a 15-bit cyclic redundancy checksum (CRC) field is incorporated after the data field. The transmitter utilizes a one-bit acknowledgement (ACK) field to receive acknowledgements from any receiver. One Delimiter each is placed after the CRC and ACK field. The message frame is terminated by a seven-bit end-of-frame (EOF) signal. In addition, the CAN protocol supports an extended data frame format with a 29-bit identifier, allowing for a larger address space. Fig. 2 illustrates the standard 11-bit identifier format.

CAN arbitration is another significant process that determines the allocation of the bus for data transmission among multiple CAN controllers. It plays a crucial role in determining the actual available bandwidth for communication [50], [51]. When a CAN controller detects an idle bus, it can initiate a transmission. The CAN protocol ensures a non-destructive bus arbitration mechanism. If all nodes on the bus transmit a recessive level, the bus remains in a recessive state, whereas if any node transmits a dominant level, the bus switches to a dominant state [52], [53]. When multiple devices attempt to transmit simultaneously, conflicts are resolved using bit-wise arbitration and the identifier of each unit [4]. The arbitration field contains the identifier, and its numerical value determines the priority of the message. Lower numerical values indicate higher priority, and the message with the lowest identifier takes precedence [54]. Fig. 3 depicts an illustrative scenario of CAN bus arbitration in an In-Vehicle Network. The scenario consists of three nodes: Node 1, Node 2, and Node 3, all contending for bus access simultaneously. The arbitration process initiates after the first Start of Frame (SOF)

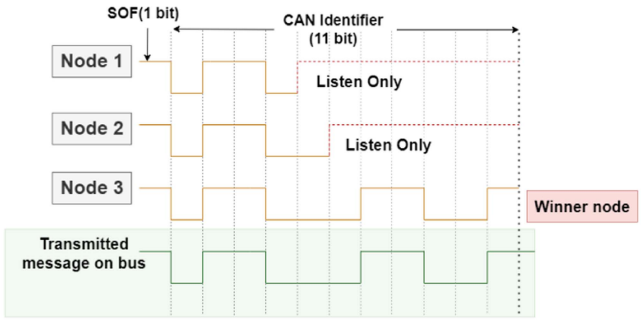


Fig. 3. CAN bus arbitration.

bit, which is dominant for all frames. During the arbitration, the frames or messages from the nodes compete for access by undergoing a bit-by-bit comparison. Each bit of the frames is evaluated, and the frame with a low bit (zero bit) emerges as the winner of arbitration. In the event of a tie, the comparison proceeds to the next bit of all three frames. This iterative process continues until a winner is determined, as each frame possesses a unique CAN identifier. In the provided example, Node 3 triumphs in the arbitration, signifying that its message holds the highest priority among the three nodes. Once the winner is established, the other nodes (Node 1 and Node 2) transition into a listen-only mode, relinquishing control of the bus. The winning node, Node 3, in this instance, gains exclusive access to the bus and can proceed with transmitting its message. This prioritization mechanism enables CAN to effectively manage and ascertain the precedence of messages on the bus.

IV. Proposed Intrusion Detection Architectures

In this research work, we present a novel two-step intrusion detection system (IDS) tailored specifically for the Controller Area Network (CAN). The overall design of our proposed system is visualized in Fig. 4. Our IDS is composed of two distinctive stages, each fulfilling a specific role in the comprehensive detection process.

The first stage of our proposed model utilizes a Long Short-Term Memory (LSTM) based architecture. This LSTM-based classifier is responsible for detecting the presence or absence of attacks within the CAN network by analyzing network data and identifying patterns indicative of potential intrusions. Once an attack is successfully detected in the initial stage, the second stage of HybridSecNet comes into action. This stage employs a Convolutional Neural Network (CNN) based multiclass classifier that focuses on precision. Its role is to accurately predict the specific type of attack encountered, categorizing the detected attack into one of four predefined classes. The combination of the LSTM-based classifier and the CNN-based multiclass classifier enables effective identification and classification of attacks along with reliability, contributing to the overall security enhancement of the CAN network and mitigating potential threats.

A. Long Short-Term Memory (LSTM) Based Classifier

In the first step of our proposed approach, we developed an LSTM-based model as a classifier to differentiate between normal and attacked behaviours within the data. The decision to employ LSTM was motivated by its capability to overcome the limitations of conventional Recurrent Neural Networks (RNNs) in capturing long-term relationships within time series data [55]. The LSTM model comprises a cell state that functions as memory, retaining information from past instances, and a "forget gate" that determines which memories can be disregarded by the cell state. These mechanisms enable the preservation of long-term dependencies between past and future instances, effectively addressing the vanishing gradient problem encountered in traditional RNNs. By leveraging the LSTM model, our system can effectively detect various attack types, whether pattern-based or frequency-based, by capturing long-term temporal trends in the attack-free data and discerning correlations among different variables in the in-vehicle network data. For training the LSTM model, the input data undergoes preprocessing using a technique called window sliding. A window size of 'n' is selected, where the initial 'n' rows from the input and output matrices are utilized to form the first entry in the transformed input and

Algorithm 1: Transform Predicted Data.

Procedure TRANSFORMDATA$y_{\text{predicted}}$
Set threshold $t = 0.5$
for each element y in $y_{\text{predicted}}$ **do**
  if $y > t$ **then**
    $y = 1$
  else
    $y = 0$
  end if
end for
Reshape $y_{\text{predicted}}$ into a 2D array with 10 rows:
  $f =$ number of elements in $y_{\text{predicted}}/10$
  $arr = \text{reshape}(y_{\text{predicted}}, (10, f))$
Initialize a new array arr_{new} with zeros, with a length of $f + 10$:
  $arr_{\text{new}} = \text{create\_array}(f + 10)$
  $\text{fill}(arr_{\text{new}}, 0)$
Get the actual array of predicted values for one-to-one testing:
for j in range(0, columns(arr) - 1) **do**
  for i in range(0, 9) **do**
    $k = i + j$
    $arr_{\text{new}}[k+1] = arr[i][j] + arr_{\text{new}}[k+1]$
  end for
end for
Each element of arr_{new} has a value between 0 and 10.
  Set a second threshold:
for i in range(0, size(arr_{new}) - 1) **do**
  if $arr_{\text{new}}[i] > 4$ **then**
    $arr_{\text{new}}[i] = 1$
  else
    $arr_{\text{new}}[i] = 0$
  end if
end for
end procedure

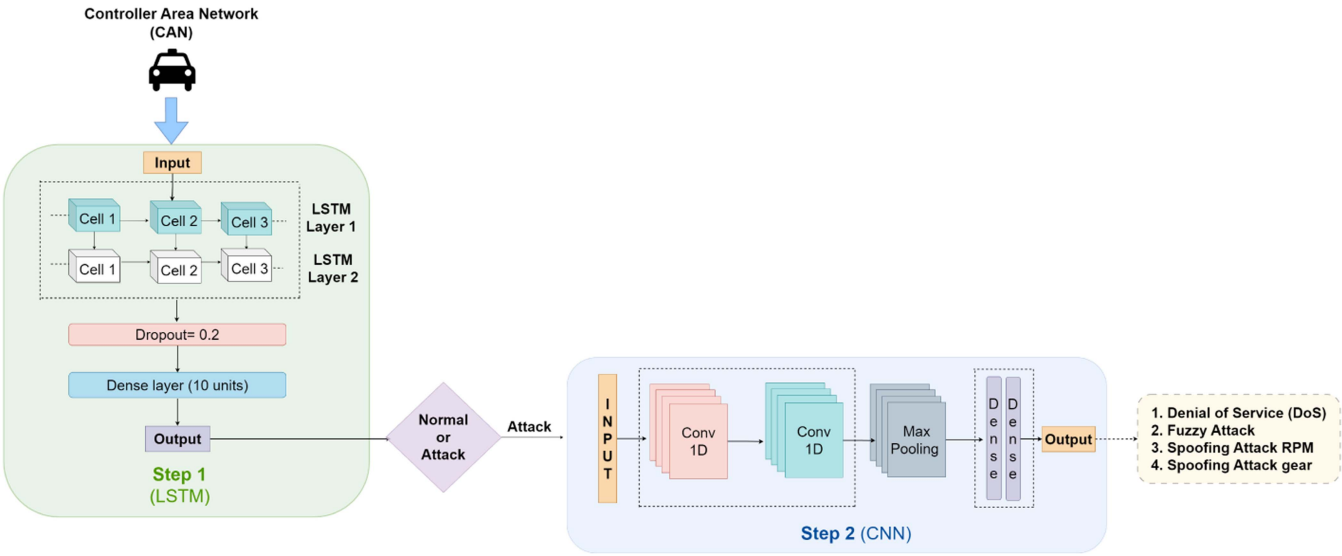


Fig. 4. Proposed two-step architecture.

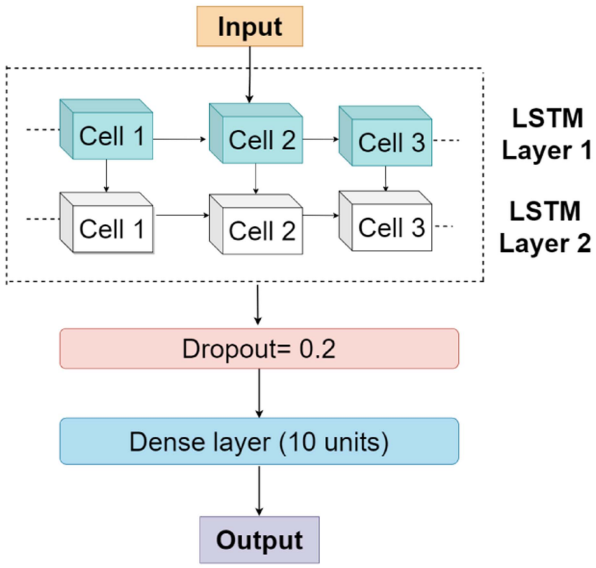


Fig. 5. Step one architecture (Long Short-Term Memory (LSTM)).

output matrices. This process is iteratively applied to populate subsequent entries in the transformed matrices. We opted for a window size of 10 when building the model. Fig. 5 represents the proposed LSTM-based step one architecture.

After preprocessing the data, the LSTM classifier is trained using a double Stacked LSTM architecture, consisting of 64 LSTM units in the first layer and 32 units in the second layer. Through experimentation, a batch size of 64 was determined to yield superior detection accuracy. The final layer of the model is a dense layer with a shape of [n*1], which receives input from the preceding LSTM layer. The hyperparameter values were carefully chosen through rigorous experimentation. The Adam optimizer with a learning rate of 0.0001 was employed, and the mean squared error (MSE) was utilized as the loss function, with the output activation function set as a sigmoid. The training process comprises 10 epochs. Once the trained model makes predictions on the transformed dataset, the inverse transformation is applied to obtain the results in their original shape. Algorithm 1 is used for the inverse transformation, followed by the establishment of a threshold to achieve optimal performance evaluation.

After obtaining predictions from our trained model, directly comparing the predicted values with the desired values is not feasible due to the window-sliding approach used for the input data, as explained above. This approach results in multiple instances of outputs corresponding to each input. To overcome this limitation, Algorithm 1 consolidates the multiple output instances into a single instance. By setting an appropriate threshold, we obtain this single output instance, allowing for a one-to-one comparison between the predicted values and the actual values to evaluate the performance of HybridSecNet.

B. Attack Categorization Using CNN

After the LSTM classifier detects malicious or abnormal data in the first step, it forwards the data to the second step for further analysis and categorization. In the second step, the attacks identified by the first step are classified into four types: DoS, Fuzzy, Gear, and RPM. Since the first step has already processed the attack-free messages, the objective of the second step is to develop a classifier capable of distinguishing between these four different attack types. To achieve this, we employ a CNN-based multiclass classifier, as the first step has already considered the temporal correlations, and now we focus on analyzing the spatial features of the attacks. Fig. 6 represents the proposed CNN-based step two architecture.

Convolutional Neural Networks (CNNs) are well-suited for detecting intricate patterns and correlations that might elude traditional machine-learning approaches. Through convolutional

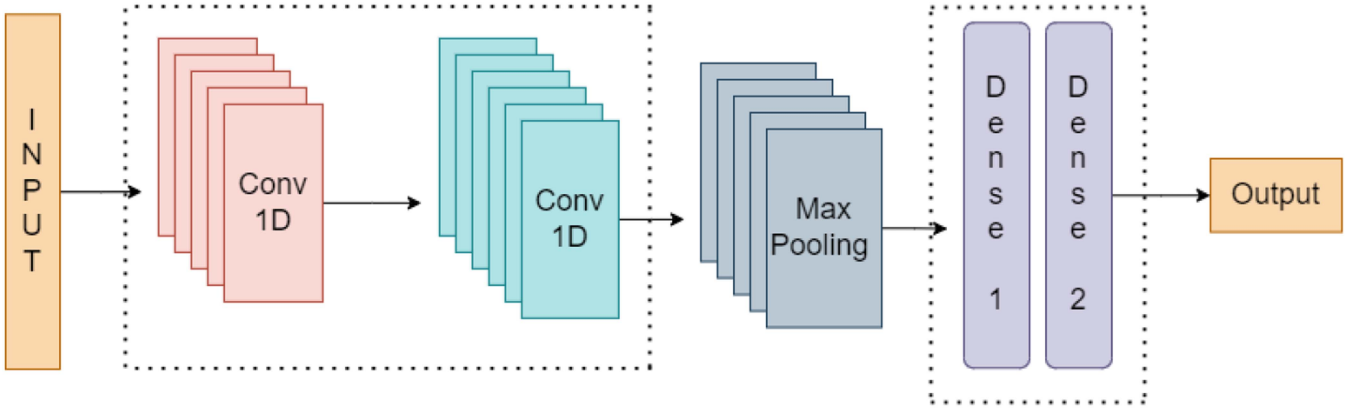


Fig. 6. Step two architecture (CNN).

operations and pooling layers, CNNs acquire hierarchical representations that capture significant spatial elements contributing to overall patterns and relationships within the dataset. To facilitate the feature generation process, we implement a function that creates features, which we further use as input to our CNN model. This function takes the preprocessed data as input, along with the corresponding labels and parameters defining the size and shape of the features. Initially, the function converts the hexadecimal values in the dataset to integers and scales specific columns using a MinMaxScaler to ensure an appropriate data format for further processing. Subsequently, the function iterates over the dataset, selecting a consecutive portion of rows as a feature and storing these features in a list. Finally, the function creates a data frame containing the generated features and their respective labels. CNN model consists of Sequential, Conv1D, and Dense layers. HybridSecNet accepts the 1D features generated by our function as input data. The model extracts relevant features using two Conv1D layers with Rectified Linear Unit (ReLU) activation, followed by a MaxPooling1D layer for downsampling. The Flatten layer is employed to prepare the data for the subsequent fully connected layers. The model comprises two Dense layers with ReLU and softmax activations, respectively, for learning and classifying high-level features. The model is trained using the Adam optimizer, categorical cross-entropy loss function, and categorical accuracy as the evaluation metric.

V. Simulation Environment

A. Dataset and Pre-Processing

HybridSecNet's training and evaluation were conducted using the Car Hacking Dataset [38]. This dataset was compiled by capturing CAN traffic from an OBD-II port of an actual vehicle while subjecting it to message injection attacks. Each dataset in this collection contains 300 instances of message injection intrusions, with each intrusion lasting between 3 to 5 seconds. The duration of CAN communication data in each dataset spans from 30 to 40 minutes. The data fields in the dataset are structured as shown in Fig. 2. The timestamp records the time in seconds, the CAN ID represents the hexadecimal identification of a CAN message (e.g., 043f), DL denotes the number of data bytes comprising the CAN frame (ranging from 0 to 8), DATA[0] contains the data value (byte) of the frame, and the Flag indicates whether the message is injected (T) or a regular message (R).

For the first step, involving the LSTM-based classifier, the CSV files corresponding to the four attack types are concatenated. The raw CAN bus dataset is initially in hexadecimal format. To train our machine learning model, we convert the hexadecimal values to decimal format. In the label column, we replace 'R' with 0 to represent normal messages and 'T' with 1 to indicate attack/malicious messages. Out of the twelve available features, we utilize ten features from the dataset: CAN ID, Data [D0-D7], and the Label column. Each byte of CAN data is assigned to a unique column (D0-D7). We omit the Timestamp element as we do not rely on time interval analysis for intrusion detection. We standardize all the data fields and separate the label column as the output variable. The dataset encompasses four attack types, including Denial of Service (DoS), Fuzzy Attack, Spoofing Attack RPM, and Spoofing Attack gear, which are explored further as,

1) *Denial of Service (DoS) attack:* The DoS attack is characterized by the transmission of high-priority messages, leading to disruption in the transmission of lower-priority signals. This kind of attack aims to increase bus utilization, causing congestion and delays in message transmission, which can have adverse effects on the arbitration process.
2) *Fuzzy attack:* A fuzzy attack, also known as a bit flipping attack, involves manipulating specific bits within a CAN message. These attacks can be difficult to detect because the erroneous data is often in close proximity to valid data, making it appear as a minor error rather than an intentional attack. Fuzzy attacks can corrupt message content, including identifiers, which can influence the arbitration process and lead to incorrect message prioritization.
3) *Gear attack:* The gear attack specifically targets messages related to the vehicle's transmission or gear-shifting system. By manipulating these messages, the attack aims to induce abnormal behaviour or complete failure in the targeted Electronic Control Units (ECUs). Such attacks can lead to hazardous situations, including abrupt gear

changes, loss of vehicle control, or mechanical damage, posing significant risks to both the driver and the vehicle's operation.

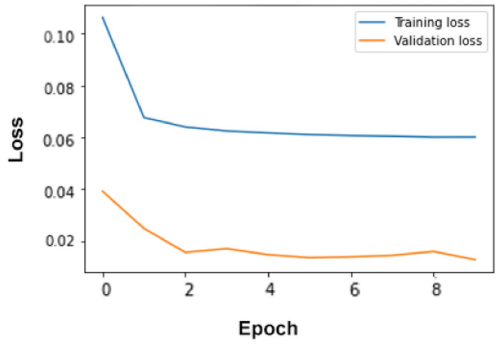
4) *RPM attack:* The RPM attack focuses on manipulating messages exchanged between Electronic Control Units (ECUs) responsible for reading RPM sensor data and displaying it on the vehicle's instrument cluster. By tampering with these messages, the attack aims to deceive or disrupt the RPM monitoring system, leading to inaccurate RPM readings being displayed to the driver. This can cause confusion and misinterpretation of the vehicle's engine speed, potentially compromising the driver's ability to make informed driving decisions and posing safety risks on the road.

VI. Results and Analysis

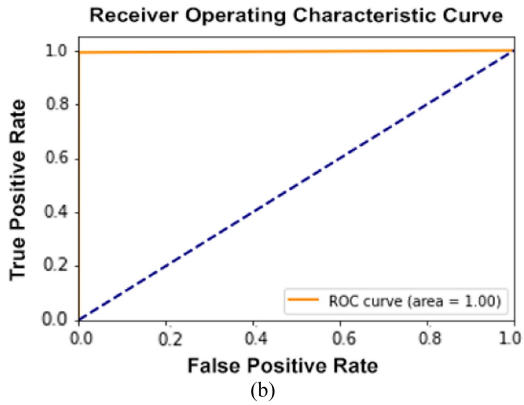
In this section, we present the performance evaluation of our proposed two-Step Intrusion Detection System (IDS) developed for modern In-Vehicle Networks (IVNs) to safeguard against four significant types of attacks. The first step of HybridSecNet involves an LSTM-based binary classifier responsible for classifying CAN messages into either normal or attack instances. Upon successful detection of an attack, the second step employs a CNN-based "4-attack classifier" to further classify the attack into one of the four attack types. Therefore, our two-step model not only predicts the presence of an attack but also identifies the specific type of attack if detected. We evaluate the performance of each step using standard evaluation metrics such as accuracy, precision, recall, and F1 score, which provide insights into the effectiveness of the model in accurately detecting and classifying attacks. The use of a binary classifier in the first step enables the training of the second step on a dataset containing only attack instances. This approach enhances the performance of the subsequent multi-class classifier by focusing exclusively on attack patterns and eliminating the complexities introduced by normal messages. As a result, the second step, the CNN-based multi-class classifier, benefits from an improved dataset, leading to enhanced classification accuracy and robustness. The numeric and graphical results of our proposed 2-Step IDS are presented, demonstrating its effectiveness in protecting IVNs from various attack types.

In Fig. 7(a), we present the variation of losses with epochs for the first step of HybridSecNet. Additionally, Graph 7(b) displays the Receiver Operating Characteristic (ROC) graph for step one, illustrating the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The area under the ROC curve, which is equal to '1' in our case, indicates nearly perfect classification.

For step two of HybridSecNet, Fig. 8 depicts the variation of Accuracy, F1-score, Loss, Precision, and Recall with epochs. The combined performance of our hybrid two-step LSTM-CNN Model is determined by multiplying the scores from both the LSTM binary classifier and the CNN multi-class classifier. This novel approach ensures that the overall accuracy, precision, recall, and F1 score are calculated based on the combined strength of both steps, as shown in Table I. Our model achieved an outstanding overall accuracy of 99.57 % and recall and precision of 99.58 % and 99.57 %, respectively.



(a)



(b)

Fig. 7. Evaluation results of step one (LSTM) (a) Loss. (b) ROC.

TABLE I
Overall Results of the Proposed 2-Step IDS Model

Metrics	Step-1	Step-2	Overall
Accuracy	99.9633	99.6316	99.595
Precision	99.9485	99.6347	99.5835
Recall	99.9481	99.6316	99.5799
F1-Score	99.9482	99.6316	99.5801

In our study, we conducted a comparative analysis between two variations of our proposed intrusion detection system: a CNN-based single-step classifier and a CNN with LSTM-based two-step classifier. The comparison aimed to evaluate the impact of the additional LSTM-based classifier in terms of classification accuracy and system efficiency. Table II presents the results of this comparison. The findings clearly demonstrate that incorporating the extra step in our IDS, utilizing the LSTM-based classifier, yields significant improvements in both classification accuracy and system efficiency. By introducing the two-step approach, we enhance the overall accuracy of our intrusion detection system, ensuring more precise and reliable identification of attacks within the CAN network. This comparison highlights the efficacy of our proposed two-step model in enhancing the performance of the intrusion detection system. The inclusion of the LSTM-based binary classifier complements the CNN-based

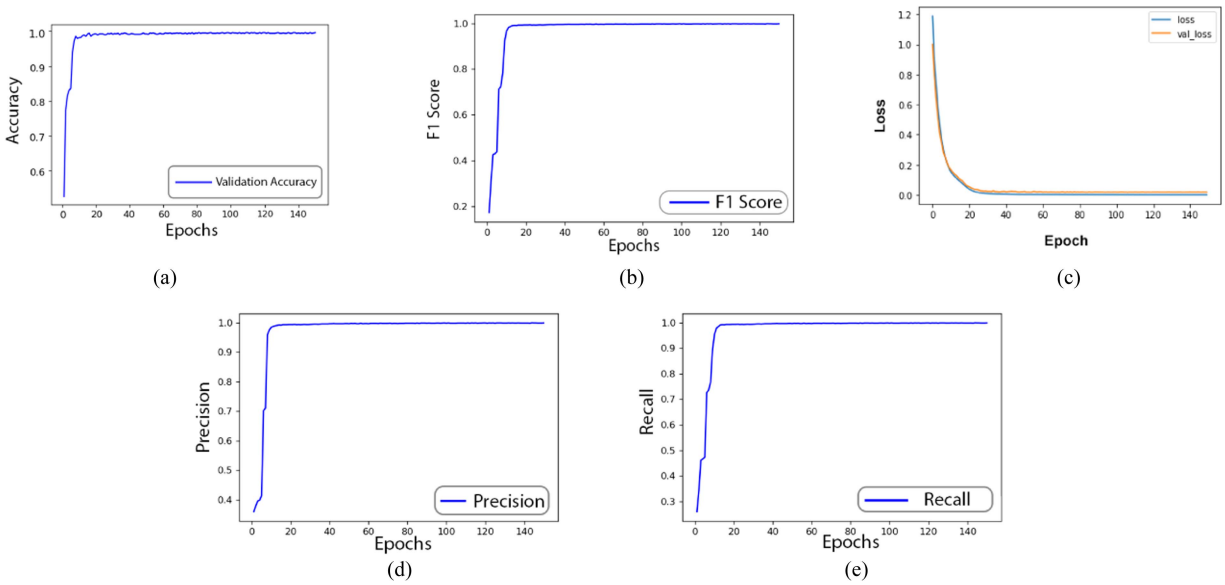


Fig. 8. Evaluation results of step two (LSTM + CNN). (a) Accuracy. (b) F1-Score. (c) Loss. (d) Precision. (e) Recall.

TABLE II
MODEL COMPARISON OF CNN-BASED MODEL AND CNN WITH LSTM MODEL

Model	Accuracy	Precision	Recall	F1 Score
Single step proposed model(CNN)	97.84	97.67	97.78	97.86
Two step proposed model(LSTM+CNN)	**99.579**	**99.583**	**99.579**	**99.580**

TABLE III
MODEL COMPARISON WITH RESPECT TO EXISTING MODELS

Model	Accuracy	Precision	Recall	F1 Score
NB	83	86	74	76
SVM	90	91	88	89
ANN	92	93	91	92
DT	92	94	78	85
BMFT	89	97	67	78
2-Stage Homogeneous [56]	97.43-99.48	97.44-99.06	97.32-99.21	96.05-99.21
2-Stage Heterogeneous [56]	95.43-98.12	94.19-99.22	95.45-98.23	92.18-98.15
Multi-Stage [57]	99.11	99.13	98.42	99.09
Two-step model (Ours)	**99.579**	**99.583**	**99.579**	**99.580**

multiclass classifier, resulting in a more robust and effective intrusion detection system that outperforms the single-step model in terms of accuracy and efficiency.

Table III compares our proposed 2-step model with other multi-step models built using the same dataset, demonstrating the superiority of our hybrid two-step LSTM-CNN model over existing multi-step models. To further validate our model's performance, we compared it against other two-step models that also utilize the car hacking dataset, using accuracy, precision, recall, and F1 score as evaluation metrics. Our model outperformed both the 2-Stage model proposed by [56] and the Multi-Stage model presented by [57], as represented in Table III. The method [56] utilizes a Rule Extraction-based architecture. Rule extraction is dependent on rules, which is not as effective as compared to LSTM. In intrusion detection, where continuous data is generated for each time span or second, representing time-based or sequential data, for such data, LSTM is well-suited over Rule extraction due to its unique architecture. Rule extraction involves simplifying the model into a set of rules, potentially leading to information loss, whereas LSTMs can retain a richer representation of data, especially in tasks where fine-grained details matter, such as in our scenario. Similarly, method [57] utilizes a bloom filter and LSTM architecture combination for malicious activities detection. The bloom filter has various limitations. Bloom filters are probabilistic data structures, meaning they can provide false positives but no false negatives. In an attack classification task, false positives could lead to incorrect predictions, and the probabilistic nature of Bloom filters might not be suitable for tasks where high precision is crucial. Additionally, bloom filters are primarily designed for set membership

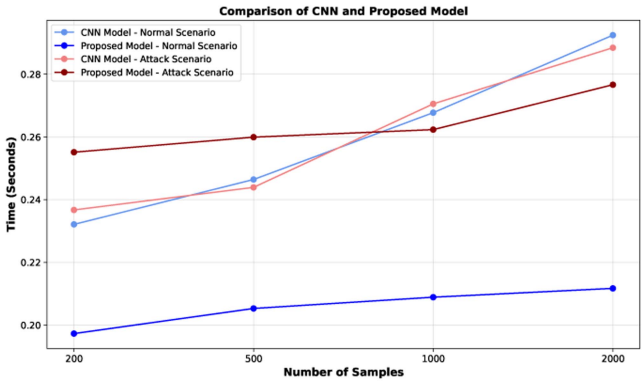


Fig. 9. Model comparison with respect to time (Seconds) and number of input samples.

testing (checking whether an element is a member of a set). They are not inherently suited for more complex classification tasks that involve distinguishing between multiple classes or categories. In contrast, our proposed CNN-based classifier (Step two in our proposed model) overcomes these limitations and provides some benefits such as multi-class classification for complex data, more in-depth and useful feature extraction, which increases the model's performance and reliability. Additionally, the method [57] is not focusing on efficient resource utilization, whereas our proposed mechanism focuses on minimal and efficient resource utilization. Based on the advantages of our proposed model architecture and the performance represented in Table III, our mechanism showcases its superiority and effectiveness in detecting and classifying attacks in IVNs.

Our investigation delves into a meticulous examination of prediction time requirements concerning the behavior of input samples, as illustrated in Fig. 9. The normal scenario, characterized by the absence of any vehicular attacks, serves as the baseline for our analysis. Our proposed model consistently exhibits superior time efficiency compared to the CNN-based model across all instances of the normal scenario. Remarkably, as the number of input samples increases, our model consistently demands less time for prediction, showcasing the efficacy of our unique model architecture design. Notably, our model avoids invoking the second step in the absence of an attack in the normal scenario, contributing to its enhanced efficiency and speed. Conversely, the CNN-based five-class classifier, designed to handle both normal and various attack types, necessitates complete processing for every input sample, resulting in higher prediction times.

Furthermore, we extend our analysis to the attack scenario, encompassing situations where attacks, such as DoS, Fuzzy, Gear, and RPM attacks, occur within the vehicle. The attack scenario amalgamates both normal and attack input samples. Initially, with a lower number of samples, the CNN-based five-class classifier demonstrates a shorter prediction time compared to our proposed model. However, as the number of samples increases, a pivotal shift occurs, and our model surpasses the CNN-based classifier, exhibiting reduced prediction times for attack scenarios. In summary, our proposed model consistently outperforms the CNN-based classifier as the number of samples increases, presenting a more favorable performance in real-world scenarios characterized by a higher volume of input samples.

VII. Conclusion

This research presents HybridSecNet, a hybrid two-step LSTM-CNN Model for Intrusion Detection, a novel deep learning-based architecture designed to enhance in-vehicle security on Controller Area Networks (CAN). The model incorporates two stages of classification, the first stage employs long short-term memory (LSTM) to classify input as either attack or normal, and the second stage further categorizes the attacks into specific types using Convolutional Neural Networks (CNN). The two-step approach significantly improves classification accuracy and reliability, achieving impressive accuracy, precision, recall, and F1-score of approximately 99.5% for CAN bus network attacks. Comparative analyses with existing single-step models highlight the superiority of our proposed model, showcasing its potential to revolutionize intelligent vehicle security and establish a robust defence against various threats in the dynamic landscape of connected vehicles. The hybrid two-step LSTM-CNN Model serves as a crucial foundation for safeguarding vehicles and paving the way for a secure and trustworthy future of in-vehicle communication and transportation.

References

[1] L. L. Bello, R. Mariani, S. Mubeen, and S. Saponara, "Recent advances and trends in on-board embedded and networked automotive systems," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 1038–1051, Feb. 2019.

[2] G. Sun, Y. Zhang, H. Yu, X. Du, and M. Guizani, "Intersection fog-based distributed routing for V2V communication in urban vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 6, pp. 2409–2426, Jun. 2020.

[3] G. Sun, L. Song, H. Yu, V. Chang, X. Du, and M. Guizani, "V2V routing in a VANET based on the autoregressive integrated moving average model," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 908–922, Jan. 2019.

[4] K. H. Johansson, M. Törngren, and L. Nielsen, "Vehicle applications of controller area network," in *Handbook of Networked and Embedded Control Systems*, Berlin, Germany: Springer, 2005, pp. 741–765.

[5] J. Lu and C. Osorio, "On the analytical probabilistic modeling of flow transmission across nodes in transportation networks," *Transp. Res. Rec.*, vol. 2676, no. 12, pp. 209–225, 2022.

[6] J. Chen, Q. Wang, W. Peng, H. Xu, X. Li, and W. Xu, "Disparity-based multiscale fusion network for transportation detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 18855–18863, Oct. 2022.

[7] Z. Fang et al., "Authority allocation strategy for shared steering control considering human-machine mutual trust level," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 2002–2015, Jan. 2024.

[8] X. Xu, W. Liu, and L. Yu, "Trajectory prediction for heterogeneous traffic-agents using knowledge correction data-driven model," *Inf. Sci.*, vol. 608, pp. 375–391, 2022.

[9] B. Xu and Y. Guo, "A novel DVL calibration method based on robust invariant extended Kalman filter," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 9422–9434, Sep. 2022.

[10] Y. Mao, Y. Zhu, Z. Tang, and Z. Chen, "A novel airspace planning algorithm for cooperative target localization," *Electronics*, vol. 11, no. 18, 2022, Art. no. 2950.

[11] T.-Y. Moon, S.-H. Seo, J.-H. Kim, S.-H. Hwang, and J. W. Jeon, "Gateway system with diagnostic function for LIN, CAN and FlexRay," in *Proc. IEEE Int. Conf. Control, Autom. Syst.*, 2007, pp. 2844–2849.

[12] C. Ding, C. Li, Z. Xiong, Z. Li, and Q. Liang, "Intelligent identification of moving trajectory of autonomous vehicle based on friction nano-generator," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 14800–14812, Dec. 2023.

[13] Y. Fu, C. Li, F. R. Yu, T. H. Luan, and P. Zhao, "An incentive mechanism of incorporating supervision game for federated learning in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 14800–14812, Dec. 2023.

[14] P. Hank, S. Müller, O. Vermesan, and J. Van Den Keybus, "Automotive ethernet: In-vehicle networking and smart mobility," in *Proc. IEEE Des., Autom. Test Europe Conf. Exhib.*, 2013, pp. 1735–1739.

[15] R. Sun, Y. Dai, and Q. Cheng, "An adaptive weighting strategy for multi sensor integrated navigation in urban areas," *IEEE Internet Things J.*, vol. 10, no. 14, pp. 12777–12786, Jul. 2023.

[16] Y. Ren, Z. Lan, L. Liu, and H. Yu, "EMSIN: Enhanced multi-stream interaction network for vehicle trajectory prediction," *IEEE Trans. Fuzzy Syst.*, early access, Feb. 01, 2024 , doi: 10.1109/TFUZZ.2024.3360946.

[17] Z. Cai, X. Zhu, P. Gergondet, X. Chen, and Z. Yu, "A friction-driven strategy for agile steering wheel manipulation by humanoid robots," *Cyborg Bionic Syst.*, vol. 4, 2023, Art. no. 00 64.

[18] Z. Xiao et al., "Understanding private car aggregation effect via spatio-temporal analysis of trajectory data," *IEEE Trans. Cybern.*, vol. 53, no. 4, pp. 2346–2357, Apr. 2021.

[19] S. Ohira, A. K. Desta, I. Arai, H. Inoue, and K. Fujikawa, "Normal and malicious sliding windows similarity analysis method for fast and accurate IDS against DoS attacks on in-vehicle networks," *IEEE Access*, vol. 8, pp. 42422–42435, 2020.

[20] X. Dai et al., "A learning-based approach for vehicle-to-vehicle computation offloading," *IEEE Internet Things J.*, vol. 10, no. 8, pp. 7244–7258, Aug. 2022.

[21] Z. Xiao et al., "Predicting urban region heat via learning arrive-stay-leave behaviors of private cars," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 10, pp. 10843–10856, Oct. 2023.

[22] X. Mo, P. Chen, J. Wang, and C. Wang, "Anomaly detection of vehicle can network based on message content," in *Proc. Secur. Privacy New Comput. Environments, 2nd EAI Int. Conf.*, 2019, pp. 96–104.

[23] X. Zhao, Y. Fang, H. Min, X. Wu, W. Wang, and R. Teixeira, "Potential sources of sensor data anomalies for autonomous vehicles: An overview from road vehicle safety perspective," *Expert Syst. Appl.*, vol. 236, 2023, Art. no. 121358.

[24] H. Min et al., "Toward interpretable anomaly detection for autonomous vehicles with denoising variational transformer," *Eng. Appl. Artif. Intell.*, vol. 129, 2024, Art. no. 107601.

[25] H. J. Jo and W. Choi, "A survey of attacks on controller area networks and corresponding countermeasures," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6123–6141, Jul. 2021.

[26] Z. Qu, X. Liu, and M. Zheng, "Temporal-spatial quantum graph convolutional neural network based on Schrödinger approach for traffic congestion prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 8, pp. 8677–8686, Aug. 2023.

[27] H. He, Z. Chen, H. Liu, X. Liu, Y. Guo, and J. Li, "Practical tracking method based on best buddies similarity," *Cyborg Bionic Syst.*, vol. 4, 2023, Art. no. 0050.

[28] W. Wu et al., "A survey of intrusion detection for in-vehicle networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 919–933, Mar. 2020.

[29] J. Luo, G. Wang, G. Li, and G. Pesce, "Transport infrastructure connectivity and conflict resolution: A machine learning analysis," *Neural Comput. Appl.*, vol. 34, no. 9, pp. 6585–6601, 2022.

[30] X. Zhang et al., "Secure routing strategy based on attribute-based trust access control in social-aware networks," *J. Signal Process. Syst.*, vol. 96, pp. 1–16, 2024.

[31] L. Yang, A. Moubayed, and A. Shami, "MTH-IDS: A multitiered hybrid intrusion detection system for Internet of Vehicles," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 616–632, Jan. 2021.

[32] Y. Zhao, Y. Xun, and J. Liu, "ClockIDS: A real-time vehicle intrusion detection system based on clock skew," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 15593–15606, Sep. 2022.

[33] H. Lee, S. H. Jeong, and H. K. Kim, "OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame," in *Proc. IEEE 15th Annu. Conf. Privacy, Secur. Trust*, 2017, pp. 57–5709.

[34] H. M. Song, H. R. Kim, and H. K. Kim, "Intrusion detection system based on the analysis of time intervals of can messages for in-vehicle network," in *Proc. IEEE Int. Conf. Inf. Netw.*, 2016, pp. 63–68.

[35] D. Stabili, M. Marchetti, and M. Colajanni, "Detecting attacks to internal vehicle networks through hamming distance," in *Proc. IEEE AEIT Int. Annu. Conf.*, 2017, pp. 1–6.

[36] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PLoS One*, vol. 11, no. 6, 2016, Art. no. e0155781.

[37] S. Tariq, S. Lee, and S. S. Woo, "CANTransfer: Transfer learning based intrusion detection on a controller area network using convolutional LSTM network," in *Proc. 35th Annu. ACM Symp. Appl. Comput.*, 2020, pp. 1048–1055.

[38] E. Seo, H. M. Song, and H. K. Kim, "GIDS: GAN based intrusion detection system for in-vehicle network," in *Proc. IEEE 16th Annu. Conf. Privacy, Secur. Trust*, 2018, pp. 1–6.

[39] K. Agrawal, T. Alladi, A. Agrawal, V. Chamola, and A. Benslimane, "NovelADS: A novel anomaly detection system for intra-vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 22596–22606, Nov. 2022.

[40] F. Amato, L. Coppolino, F. Mercaldo, F. Moscato, R. Nardone, and A. Santone, "Can-bus attack detection with deep learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5081–5090, Aug. 2021.

[41] T. Alladi, B. Gera, A. Agrawal, V. Chamola, and F. R. Yu, "DeepADV: A deep neural network framework for anomaly detection in VANETs," *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 12013–12023, Nov. 2021.

[42] M. H. Shahriar, Y. Xiao, P. Moriano, W. Lou, and Y. T. Hou, "CANShield: Deep learning-based intrusion detection framework for controller area networks at the signal-level," *IEEE Internet Things J.*, vol. 10, no. 24, pp. 22111–22127, Dec. 2023.

[43] W. Jeong, S. Han, E. Choi, S. Lee, and J.-W. Choi, "CNN-based adaptive source node identifier for controller area network (CAN)," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13916–13920, Nov. 2020.

[44] A. R. Javed, S. Ur Rehman, M. U. Khan, M. Alazab, and T. Reddy, "CANintelliIDS: Detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1456–1466, Feb. 2021.

[45] H. Zhang, X. Meng, X. Zhang, and Z. Liu, "CANsec: A practical in-vehicle controller area network security evaluation tool," *Sensors*, vol. 20, no. 17, 2020, Art. no. 4900.

[46] Aptiv, "What is an electronic control unit? Aptiv," 2023. [Online]. Available: https://www.aptiv.com/en/insights/article/what-is-an-electronic-control-unit

[47] K. Cheng, Y. Bai, Y. Zhou, Y. Tang, D. Sanan, and Y. Liu, "CANeleon: Protecting CAN bus with frame ID chameleon," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7116–7130, Jul. 2020.

[48] A. Mohammadzadeh, H. Taghavifar, C. Zhang, K. A. Alattas, J. Liu, and M. T. Vu, "A non-linear fractional-order type-3 fuzzy control for enhanced path-tracking performance of autonomous cars," *IET Control Theory Appl.*, vol. 18, no. 1, pp. 40–54, 2024.

[49] R. Luo, Z. Peng, J. Hu, and B. K. Ghosh, "Adaptive optimal control of affine nonlinear systems via identifier–critic neural network approximation with relaxed pe conditions," *Neural Netw.*, vol. 167, pp. 588–600, 2023.

[50] J. Zhao, D. Song, B. Zhu, Z. Sun, J. Han, and Y. Sun, "A human-like trajectory planning method on a curve based on the driver preview mechanism," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 11, pp. 11682–11698, Nov. 2023.

[51] R. Wang et al., "FI-NPI: Exploring optimal control in parallel platform systems," *Electronics*, vol. 13, no. 7, 2024, Art. no. 1168.

[52] S. K. R. Gurram, "Implementation of controller area network (CAN) bus in an autonomnous all-terrain vehicle," Ph.D. dissertation, The University of North Carolina at Charlotte, Charlotte, NC, USA, 2011.

[53] W. Wang, J. Liang, M. Liu, L. Ding, and H. Zeng, "Novel robust stability criteria for Lur'e systems with time-varying delay," *Mathematics*, vol. 12, no. 4, pp. 630–642, 2024.

[54] J. Cook and J. Freudenberg, "Controller area network (CAN)," *EECS*, vol. 461, pp. 1–5, 2007.

[55] Z. Khan, M. Chowdhury, M. Islam, C.-Y. Huang, and M. Rahman, "Long short-term memory neural network-based attack detection model for in-vehicle network security," *IEEE Sensors Lett.*, vol. 4, no. 6, pp. 1–4, Jun. 2020.

[56] S. Almutlaq, A. Derhab, M. M. Hassan, and K. Kaur, "Two-stage intrusion detection system in intelligent transportation systems using rule extraction methods from deep neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 15687–15701, Dec. 2023.

[57] I. A. Khan, N. Moustafa, D. Pi, W. Haider, B. Li, and A. Jolfaei, "An enhanced multi-stage deep learning framework for detecting malicious activities from autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25469–25478, Dec. 2021.

Amit Chougule received the M.Tech. degree from PES University, Bangalore, India, in 2020, and the Ph.D. degree in computer vision and AI from the Birla Institute of Technology And Science - Pilani, Pilani, India, in 2024. In 2023, he was a Visiting Researcher with the Trust in Connected and Autonomous Vehicles (TrustCAV) Research Group, Carleton University, Ottawa, ON, Canada. Throughout his career, he has held significant roles as an artificial intelligence and medical imaging Researcher within esteemed organizations such as Sony Research, Philips Healthcare, and AIvolved Technologies. His research interests include the development of artificial intelligence solutions for autonomous driving and healthcare applications, leveraging his expertise in computer vision and deep learning methodologies.

Ishan Kulkarni received the Bachelor of Engineering degree from the Department of Electrical & Electronics Engineering, Birla Institute of Technology And Science - Pilani, Pilani, India. He is actively conducting research in the domain of networking and security for connected cars. He is currently a Software Developer with Rigi, Bengaluru, India. His research interests include security for autonomous driving, CAN, machine learning, and computer vision for self-driving cars.

Tejasvi Alladi (Senior Member, IEEE) received the B.E. degree from the Birla Institute of Technology and Science Pilani (BITS-Pilani), Pilani, India, in 2010, the M.S. degree from North Carolina State University, Raleigh, NC, USA, in 2015, and the Ph.D. degree from BITS-Pilani, in 2021. From January 2021 to December 2021, he was a Postdoctoral Researcher with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada. He is currently an Assistant Professor with the Department of Computer Science and Information Systems, BITS-Pilani. He has around six years of industrial experience working on embedded systems in semiconductor MNCs, such as qualcomm technologies and samsung electronics. His research interests include developing security solutions for the Internet of Things using cryptography, deep learning, and blockchain technologies.

Vinay Chamola (Senior Member, IEEE) received the B.E. and M.E. degrees from the Birla Institute of Technology And Science - Pilani (BITS-Pilani), Pilani, India, in 2010 and 2013, respectively, and the Ph.D. degree from the National University of Singapore, Singapore, in 2016. He is currently an Associate Professor with the Electrical and Electronics Department, BITS-Pilani and is also a part of APPCAIR, BITS-Pilani. He has more than 100 publications in high-ranked SCI journals, including more than 75 IEEE transactions, journal, and magazine articles. His research interests include the Internet of Things, 5G network provisioning, blockchain, and security. He is an Area Editor of *Ad Hoc Networks*, Elsevier, and *IEEE Internet of Things Magazine*. He is also an Associate Editor for various journals, including IEEE NETWORKING LETTERS, *IEEE Consumer Electronics Magazine*, *IET Networks, and IET Quantum Communications*.

Fei Richard Yu (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of British Columbia, Vancouver, BC, Canada, in 2003. From 2002 to 2006, he was with Ericsson, Lund, Sweden, and a start-up in San Diego, CA, USA, where he worked on the research and development in the areas of advanced wireless communication technologies and new standards. He joined Carleton School of Information Technology and the Department of Systems and Computer Engineering (cross-appointment), Carleton University, Ottawa, ON, Canada, in 2007, where he is currently a Professor. He has authored or coauthored more than 600 papers in reputable journals/conferences, eight books, and 28 granted patents, with more than 10 000 citations (Google Scholar). His research interests include cyber-security, connected and autonomous vehicles, artificial intelligence, blockchain, and wireless systems. He was the recipient of the Distinguished Service Awards in 2019 and 2016, Outstanding Leadership Award in 2013, Carleton Research Achievement Awards in 2012 and 2021, Ontario Early Researcher Award (formerly Premiers Research Excellence Award) in 2011, Excellent Contribution Award at IEEE/IFIP TrustCom 2010, and Leadership Opportunity Fund Award from Canada Foundation of Innovation in 2009.