IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 7 October 2023, accepted 24 October 2023, date of publication 27 October 2023, date of current version 3 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3328182

RESEARCH ARTICLE

Transfer Learning-Based Intrusion Detection System for a Controller Area Network

NARAYAN KHATRI[1], SIHYUNG LEE[2], AND SEUNG YEOB NAM[1], (Senior Member, IEEE)
[1]Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, South Korea
[2]School of Computer Science and Engineering, Kyungpook National University, Daegu 41566, South Korea

Corresponding author: Seung Yeob Nam (synam@ynu.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) funded by the Korean Government [Ministry of Science and ICT (MSIT)] under Grant 2020R1A2C1010366, and in part by the Basic Science Research Program through the NRF funded by the Ministry of Education under Grant 2021R1A6A1A03039493.

ABSTRACT The Controller Area Network (CAN) is a major protocol for in-vehicle network communications. This protocol is simple and efficient for message transmission and the smooth functioning of an in-vehicle system. On the other hand, the weaknesses of this protocol, such as the ID-based arbitration method for message transmission and lack of authentication mechanism, make it vulnerable to various security attacks, including DoS attacks, Fuzzy attacks, impersonation attacks, and replay attacks. Since there is no authentication mechanism for transmitted messages, we need a way to distinguish between normal and attack messages. An intrusion detection system (IDS) is an option for this problem because it can raise alarms when there are flaws in the system. IDS is very efficient for intrusion detection where messages with the same IDs are transmitted periodically. The deviation from the normal pattern of message transmission will force the IDS system to trigger alarms. Most studies on the CAN bus IDS system were based on a supervised learning approach. On the other hand, the lack of labeled datasets and a huge amount of training time make it inefficient for new attack patterns. This paper proposes a transfer learning-based IDS system for in-vehicle network intrusion detection. The extraction of quality features using transfer learning (TL) and appropriate fine-tuning methodology is used in the proposed model. This approach can use the available intrusion attack dataset to detect new attacks. The experimental results indicated that the proposed deep hybrid transfer learning (TL) model detects new threats with a high accuracy of approximately 99.9% when compared to state-of-the-art methods, while also lowering training and testing time by more than 30%.

INDEX TERMS VANETs, intrusion detection system (IDS), transfer learning, supervised learning, security.

I. INTRODUCTION

In-vehicle networks serve as the foundation for modern automobile operation. The in-vehicle system consists of various Electronic Control Units (ECUs), such as Transmission Control Unit (TCU), Anti-lock Braking System (ABS), Body Control Module (BCM), Speed Control Unit, Powertrain Control Module (PCM), and Door Control Unit (DCU) [1]. These ECUs have their specific functions for safe control of the vehicle. These ECUs are connected with the standard protocol. Various protocols are used for in-vehicle network communications, including Controller Area Network (CAN), CAN Flexible Data-Rate (CAN FD), Ethernet, FlexRay, Local Interconnect Network (LIN), and Media Oriented Systems Transport (MOST) protocol [1].

CAN is the standard protocol used widely for the interconnection of various ECUs in in-vehicle networks. This protocol is robust with a less complex design. It is popular among automobile manufacturers because of its low design cost. Despite these advantages, security has not been considered in the design of this protocol. Therefore, this protocol can be vulnerable to security attacks. There is no information regarding the sender and receiver of the message in this protocol. Fig. 1 depicts the CAN bus protocol frame format. This protocol is a broadcast-type protocol. The CAN

The associate editor coordinating the review of this manuscript and approving it for publication was Barbara Masini.

message is broadcast throughout the network, and the receiver node can decide whether to accept or reject it based on the priority mechanism. The protocol works based on a priority mechanism. It uses the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) arbitration mechanisms [1]. The messages with the higher priority (i.e., lower id) will suppress the flow of low-priority messages. For example, the messages for PCM have higher priority over the DCU messages. This prioritization mechanism allows hackers to inject higher-prior messages to control the bus protocol. Hoppe, Miller, and Nie reported practical attacks exploiting the CAN bus vulnerability [2], [3], [4]. These authors were able to launch physical as well as remote attacks on modern vehicles. Miller and Valasek hacked Jeep Cherokee and demonstrated the vulnerability of the CAN networks by disabling the critical functionality of the vehicle [3]. The CAN bus arbitration mechanism and lack of message encryption lead to various network vulnerabilities. Various attacks, such as DoS, fuzzy, replay, spoofing, and impersonation attacks, are the common attack types on in-vehicle CAN bus networks [1], [5]. Messages exchanged between various ECUs via the CAN bus do not meet the security requirements. Unfortunately, these messages are neither authenticated nor encrypted [6].

Vehicles are becoming smarter, with connections to other vehicles and external networks established. The cellular 5G and New Radio (NR) Vehicle-to-Everything (V2X) technology (i.e., cellular 5G NR V2X) will lead the modern vehicular ad-hoc networks (VANETs) [7], [8]. This V2X communication technology can fulfill the latency, bandwidth, networking, and security requirements essential for future autonomous vehicles. The advances in this technology will lead to an increase in vehicle security attacks. A previous study [1] reported cases of attacks on these networks. A significant rise in vehicular network attack surfaces is expected as the development of smart and autonomous vehicles accelerates in the coming future [9], [10], [11]. Furthermore, issues such as trustworthiness of messages in VANET should be handled properly [12]. The injection of false messages may lead to collateral damage to a vehicle. The usage of various sensors and cameras in the vehicle results in data availability. The black box information can provide information regarding vehicle collisions [13]. We can examine these data using a data analysis technique that employs a machine learning algorithm to discover and track accidents. In case for CAN network, there is no chance of complete replacement of this protocol in the near future. Therefore, security solutions are needed to protect in-vehicular networks. Authentication and encryption-based security solutions are inefficient because many messages are generated in this network within milliseconds. An intrusion detection system (IDS) is the best option for these networks because the CAN messages follow specific patterns and any deviations from these patterns can be considered an anomaly. The report from the IDS system can be used by the network administrator to take early action.

An intrusion detection system is a software program that can detect suspicious activity in a network or a system. The IDS types can be signature-based, anomaly-based, misuse-based, and hybrid intrusion detection systems [6]. An anomaly-based intrusion detection system is suitable for CAN messages because regular CAN messages show discernible patterns that set them apart from abnormal messages, including consistent repetition of an ID at regular intervals [5] and specific arrangements among a group of IDs [14]. With this approach, the injected messages that show deviations from normal messages can be detected easily. Deep learning has been widely used for intrusion detection in CAN networks. However, deep-learning based security solutions for in-vehicle networks require a large amount of training data. Despite the security flaws, the in-vehicle dataset is not readily available for security analysis. The automotive industry can not publicize vehicle data because of security concerns and other reasons [14], [15]. The collection and labeling of CAN data are expensive and time-consuming. With technological advances, new attack patterns also appear in vehicular networks. The IDS system should be dynamic and adaptive enough to detect new intrusion messages. With conventional machine learning algorithms, the model needs to be trained whenever a new attack pattern is detected. Thus, there is a need of an algorithm that can use the features learned from previous learning algorithms and utilize it for future intrusion detection purposes. Transfer learning (TL) can be an optimal solution for intrusion detection in this network, because it has the capability of knowledge transfer and learning in a dynamic manner [16]. TL is a machine learning strategy that uses the features acquired from one task to solve another. Pre-trained models are developed on large-scale datasets from a certain domain and are used to train a new model, reducing training time and generalization error. When there is insufficient data to train a full-scale deep learning model from scratch, this strategy can be useful.

This paper provides with experiments on the feasibility of TL for intrusion detection in CAN networks. TL can be an option for mitigating the drawbacks observed when deploying traditional machine learning and deep learning-based IDS systems. For this purpose, a hybrid transfer learning-based IDS model was developed using CNN and LSTM machine learning algorithms. The effectiveness of the proposed model was tested using two sets of real-world datasets and evaluated based on several machine learning evaluation metrics.

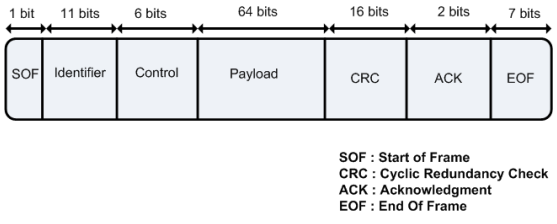

FIGURE 1. CAN bus protocol frame format.

A. CONTRIBUTIONS

The main contributions of this paper can be summarized as follows:

- This paper proposes an IDS for CAN based on deep learning. Unlike previous studies that used individual machine-learning models, such as CNN, LSTM, or GAN, the proposed system employs a hybrid approach combining CNN and LSTM. By leveraging CNN, the system effectively extracts distinctive features characterizing individual messages, while LSTM enables the correlation of these features among subsequent messages, leading to the accurate identification of a series of attack messages.
- Furthermore, a transfer-learning method specifically tailored for the hybrid model was also proposed. This approach allows the model to learn new attacks while leveraging the knowledge gained from previous attacks, substantially reducing the time and effort required to retrain the entire model from scratch.
- To validate the effectiveness of our proposed methods, we conducted evaluations using two real-world datasets containing nearly 21 million CAN messages, including four types of attack patterns. The results demonstrate that our IDS successfully detects attack messages with an accuracy exceeding 0.999, outperforming previous schemes. Furthermore, our transfer-learning approach reduces training time by more than 30% of the time required for training from scratch while maintaining a high level of accuracy.

The abbreviations used in this paper are listed in Table 1.

B. PAPER ORGANIZATION

The remainder of the paper is organized as follows. Section II outlines the related studies on machine learning-based IDS systems in CAN networks. Section III describes the proposed transfer learning-based anomaly detection system. Section IV provides experimentation and performance analysis of the proposed algorithm. Finally, the paper is concluded with future work in Section V.

II. RELATED WORK

Academia and industry are interested in developing a solution for underlying vulnerabilities in CAN Bus networks. The use of machine learning and deep learning for anomaly identification in CAN networks has attracted considerable attention. The widespread use of video cameras and sensors, such as light detection and ranging (LIDAR) sensors, radio detection and ranging (RADAR) sensors, and ultrasonic sensors, has aided the development of autonomous and self-driving vehicles. The in-vehicle networks support the exchange of sensory signals between the ECUs. These sensors generate vast amounts of data that can be used to discover anomalies using machine learning algorithms and data analysis techniques.

Avatefipour et al. proposed a modified one-class support vector machine algorithm for anomaly detection in CAN bus networks [5]. The messages in CAN networks exhibit some patterns from the repetition of a specific ID at regular intervals. The main idea is that deviations from usual message patterns are considered abnormal. The proposed algorithm is optimized and selects the best parameters for the anomaly detection model. The model consists of training and testing phases. During the training phase of one-class SVM, a meta-heuristic optimization approach is used to identify the appropriate kernel type and function that provides the optimal hyperplane and the ideal support vectors.

Hanselmann et al. proposed an unsupervised machine learning algorithm using long short-term memory (LSTM) networks [14]. The authors proposed a CANet architecture that captures the temporal features of each individual CAN ID with their corresponding LSTM input model. The output of all input models was aggregated and passed into a fully connected autoencoder subnetwork. This allowed the network to consider the interdependence of signals from all IDs. All potential input signals were reconstructed at each point in time. The anomaly score could be calculated using the reconstruction error between the correct signal levels and their reconstruction.

Long Short-Term Memory (LSTM) has been proposed for intrusion detection in CAN bus communications [6]. Various attacks, such as Denial of Service (DoS), fuzzy attack, and spoofing attacks for handle angle and vehicle speed, have been examined on actual Toyota hybrid cars. The authors generated real and synthetic datasets for experimentation purposes. The dataset was preprocessed and fed to the LSTM network, where suitable hyperparameter tuning was performed. The experimental results revealed the high accuracy of 99.995% and low false positive rates of the proposed algorithm.

Seo et al. proposed a Generative Adversarial Network (GAN)-based intrusion detection system that uses a deep learning model to detect unknown attacks using only normal data [17]. The proposed model consists of two discriminators and one generator. The first discriminator is trained with normal and abnormal CAN data images. In the second stage of the training process, normal CAN images and fake images produced by the generator are passed to the second discriminator. This discriminator will discriminate whether the received images are normal CAN or abnormal.

An evolutionary optimization algorithm using a deep denoising autoencoder has been developed as an anomaly detection framework [18]. The authors discuss the difficulties of premature convergence and optimal network structure selection in deep learning algorithms and propose an ecogeography-based optimization strategy for dealing with these concerns.

Deep convolutional neural networks have been studied for intrusion detection in in-vehicle networks [19]. The authors developed deep learning architecture using the Inception-ResNet model, and experimentation was performed by injecting attack messages in a real vehicle. The modified inception resnet model was designed to categorize

TABLE 1. List of abbreviations.

Abbreviation	Full Form
CAN	Controller Area Network
CAN FD	CAN Flexible Data Rate
MOST	Media-Oriented Systems Transport
IDS	Intrusion Detection System
VANET	Vehicular Ad-hoc Network
V2X	Vehicle-to-Everything
ECU	Electronic Control Unit
ABS	Anti-lock Braking System
BCM	Body Control Module
PCM	Powertrain Control Module
DCU	Door Control Unit
TCU	Transmission Control Unit
LIN	Local Interconnect Network
ANN	Artificial Neural Network
LSTM	Long Short-Term Memory
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
DNN	Deep Neural Network
DOS	Denial Of Service
GAN	Generative Adversarial Network
DL	Deep Learning
TL	Transfer Learning
OBD	On-board Diagnostics
RELU	Rectified Linear Unit
MMD	Maximum Mean Discrepancy
DLC	Data Length Code
NR	New Radio
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
LIDAR	Light Detection and Ranging
RADAR	Radio Detection and Ranging

$29 \times 29 \times 1$ input data into two classes, whereas the original inception resnet will classify $299 \times 299 \times 3$ input images into 1000 classes. Their work was the first to use a CNN-based deep learning algorithm for intrusion detection in CAN networks. They also pioneere in building a labeled in-vehicle network attack dataset that includes normal and attack patterns. DoS, fuzzy, and impersonation injection attacks were carried out in a real vehicle using custom-built Raspberry Pi devices connected to the in-vehicle network via the OBD-II connection in the vehicle. Extensive simulations were carried out to evaluate the proposed deep learning algorithm on the dataset they produced. Their method reduced the false negative and error rates and improved the precision, recall, and f1 scores.

Mehedi et al. suggested a transfer learning-based intrusion detection system for electric vehicles [9]. Their research focused on anomaly detection using a deep learning-based transfer learning approach with the optimal feature selection methodology. The deep learning model used was the LeCun Network (LeNet) model, which was evaluated on a real-world dataset consisting of flooding, spoofing, replay, and fuzzing attack patterns. The experimentation consisted of two phases: training and validation. The LeNet model was trained with a vast data during the training process, and the optimal model parameters that can improve learning were chosen. The model was evaluated with an unknown dataset to forecast the occurrence of an intrusion. The results showed that the LeNet model could achieve 98.10% accuracy, which is higher than inception and resnet networks.

Tariq et al. [20] proposed transfer learning based IDS using one-shot learning. In one-shot learning, they need to train only one instance of an intrusion type. This training can be used to detect all the instances related to a particular intrusion category. Using transfer learning, the authors showed that their method could detect unknown attack types with a few new training datasets.

The feasibility study of the multi-task transfer learning for the case of the lack of labeled datasets or the case of a small number of training datasets has been performed by Otoum et al. [21]. The common features from two sets of different datasets were mapped during the preprocessing stage and passed to the deep learning algorithms. They proposed the inductive multi-task transfer learning algorithm for transferring knowledge gained from one set of datasets to the other with the same feature space.

A previous study [16] reported the disadvantages of an IDS system due to the lack of labeled datasets. The authors demonstrated with experimentation the importance of transfer learning for developing a network IDS system with fewer new labeled datasets. In contrast to traditional machine learning and deep learning algorithms, the transfer-learning approach can work with fewer labeled datasets, reducing the complexity and training time of the algorithm.

Khademi, Ebrahimi, and Kordy investigated the use of a transfer learning-based CNN and LSTM hybrid system for EEG signal classification [22]. The hybrid neural network models were created utilizing customized CNN, ResNet-50,

and Inception-v3 networks. The spatial and sequential properties of the EEG data were extracted using these models. The pre-trained network was created with ResNet-50 and Inception-v3 and the weights were frozen in order to perform transfer learning. In our paper, we use transfer learning to develop an intrusion detection system for CAN networks. To the best of our knowledge, this work is the first attempt to utilize a transfer learning-based CNN and LSTM hybrid system in this IDS field.

Sun et al. [23] developed an attention model (i.e., CLAM) for anomaly detection for CAN networks using hybrid CNN-LSTM algorithm. For each collection of messages with the identical IDs, the authors built a distinct CLAM model. The raw CAN bus sensor signals are input to the attention model consisting of one-dimensional convolution (Conv1D) layer and the bidirectional LSTM layer. These layers will extract suitable features for the CAN sensory signals and any deviations from these signals is treated as an anomaly. This model have several drawbacks. The attacker can investigate the CAN message IDs and its functionality in the control of a specific functions of a vehicle. Then, he/she can launch an attack with the injection of combined CAN messages with heterogeneous IDs. Since the CLAM model is built for each ID separately, it will fail to classify this type of attacks including several IDs. Secondly, the model is built on only 15 different IDs with suppression of several IDs due to lack of data payload. Thus, their IDS system might fail to classify attacks on the IDs, where the model has not been established. In our paper, we build an intrusion detection system using transfer learning model which works in a supervised manner. The pre-trained model for transfer learning is developed using the hybrid CNN-LSTM algorithm. The features learned through the algorithm can be utilized to detect new attacks with a high detection accuracy.

Lo et al. proposed a hybrid intrusion detection system using the combination of CNN and LSTM deep learning algorithms [24]. Their proposed HyDL-IDS model is based on supervised learning and have high detection accuracy for attack patterns in CAN networks. The authors showed that the combination of features related to space and time can be a valuable insight for sophisticated attack detection purposes. The authors claim that the detection accuracy of their approach is approximately 100% through the experimentation performed on several attacks like DoS, fuzzy, spoofing gear, and RPM attacks. Despite the performance of the model, the authors do not provide any analysis on the computational overhead of the algorithm. The training time of the algorithm is high and it may not be suitable for CAN networks with limited computational ability. Thus, our paper proposes a transfer learning based approach for IDS in CAN networks. With the use of transfer learning, we can minimize the computational time for training a deep learning algorithm and increase the efficiency of IDS system. This approach further has the ability to dynamically detect new intrusion attacks that can be generated in the future and does not require us to train the algorithm from scratch.

Table 2 lists the classification of related works based on several metrics.

III. PROPOSED ANOMALY DETECTION SYSTEM

This section contains a problem statement and a solution formulation for developing CAN bus IDS. The proposed IDS system is illustrated with explanations and diagrams as required.

A. PRELIMINARIES

This section provides the background knowledge on neural networks for developing an intrusion detection system using the hybrid cnn-lstm model and transfer learning approach. The theoretical and mathematical formulations for the CNN and LSTM models are explained.

1) CONVOLUTIONAL NEURAL NETWORK (CNN)

CNN is a deep learning algorithm that can take an input image, give value (in the form of learnable weights and bias) to numerous objects in the image, and distinguish one from another. CNNs are typically designed for image analysis tasks, such as image and video recognition, medical image analysis, and image classification. This neural network architecture transforms large input images into smaller subsets called filters. The matrix operation between the filters and the input image section will help extract meaningful features. Fig. 2 shows the CNN network architecture of LeNet. A CNN network consists of various layers, such as convolution, pooling, and fully connected layers. Multiple convolution and pooling operations can be repeated, followed by one or more fully connected layers. Convolution is a linear operation that produces the feature map by multiplying a set of weights with the input image (n-dimensional matrices). It extracts high-level features, such as edges, from an input image. The feature map is generated through the sum of the dot product between each element of the filter and the input tensor matrix to produce a convoluted feature map. This convoluted feature is then passed as input to the next layer of the CNN model. Fig. 3 presents the process for producing a convoluted feature map.

The ReLU activation function is applied after each convolution layer to provide non-linearity to the network. The activation function performs an elementwise nonlinear transformation and sets all the negative pixels to 0. This function attempts to solve the vanishing gradient problem encountered when other activations, such as sigmoid or tanh, are used. The equation for the ReLU activation function is expressed as equation (1):

$$f(x) = \begin{cases} 0, & \text{for } x < 0, \\ x, & \text{for } x \geq 0 \end{cases} \tag{1}$$

The pooling procedure, also known as downsampling, reduces the dimensionality of each feature map. Downsampling subsamples larger size feature maps to produce smaller feature maps, keeping the most dominating features. Pooling

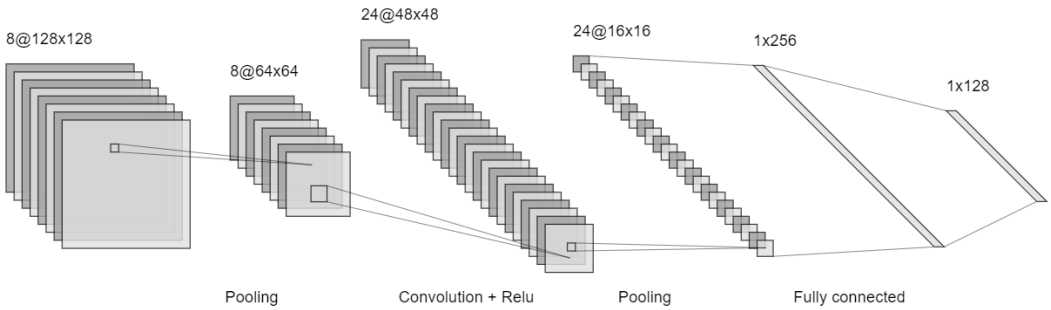

FIGURE 2. Convolutional neural network.

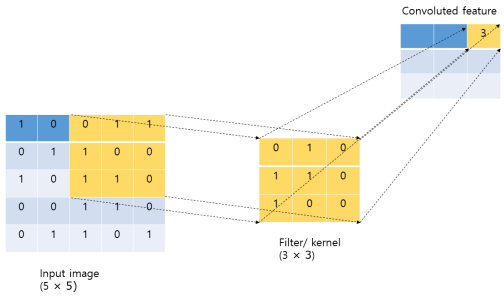

FIGURE 3. Process of convolution.

can control overfitting by reducing the number of parameters and calculations in the network. Pooling can be implemented using a variety of nonlinear functions. The most common form of pooling used in CNN is called max pooling, which divides the input image into rectangles and outputs the maximum for each sub-region.

After a series of convolution and pooling operations, there is a fully connected (FC) layer that produces the classification output of the CNN algorithm. The working mechanism of FC layers is the same as the traditional ANN, such as multi-layer perceptron. The high-level features extracted after the convolution and pooling operations are flattened to a one-dimensional vector form and are fed as input to the FC layer. The final layer of the FC layer is a classifier that classifies the input images into several classes, as in a classification problem.

2) LONG SHORT TERM MEMORY (LSTM) NETWORK

LSTM is a particular type of recurrent neural network (RNN) different from the standard feedforward neural networks. The difference is due to the feedback connection in the LSTM network, which works well for sequential and time series data [6]. Hochreiter and Schmidhuber proposed the LSTM in 1997, which can handle long-term dependencies (i.e., memorize information for more extended periods) [25]. A traditional RNN suffers from the vanishing gradient problem. LSTM was developed to solve this problem. Fig. 4 shows the architecture of the LSTM cell. A LSTM unit comprises a cell, an input gate, an output gate, and a forget gate. The cell functions as a memory, with the three gates updating and controlling the cell states. The input to the LSTM network is the input vector i_t and the hidden input vector h_{t-1}. The output is a vector h_t, as shown in Fig. 4. The forget gate decides what information to store and which information to forget from the cell state C_{t-1} based on the current input i_t and the previous cell output h_{t-1}. The sigmoid activation function is used for this purpose, which generates an output with values ranging between 0 and 1. The output 1 means to store that information, and 0 means to forget the information from the cell. The output of the forget gate f_t is computed as equation (2).

$$f_t = \sigma(W_f \times [h_{t-1}, i_t] + b_f) \tag{2}$$

where W_f and b_f denote the weights and bias for the forget gate.

The input gate decides on the new information that needs to be updated in the new cell state C_t. It can be calculated as n_t in (3).

$$n_t = \sigma(W_n \times [h_{t-1}, i_t] + b_n) \tag{3}$$

where W_n and b_n denote the weights and bias for the input gate.

The vector of the cell state is calculated using the hyperbolic tangent (tanh) function of the current input and the last hidden state, as expressed in equation (4).

$$\tilde{C}_t = tanh(W_C \times [h_{t-1}, i_t] + b_C) \tag{4}$$

where W_C and b_C denote the weights and bias for the input gate.

The updated new cell state can be calculated as equation (5).

$$C_t = f_t \times C_{t-1} + n_t \times \tilde{C}_t \tag{5}$$

The output gate activation is computed by a sigmoid function as expressed in equation (6).

$$o_t = \sigma(W_o \times [h_{t-1}, i_t] + b_o) \tag{6}$$

where W_o and b_o denote the weights and bias for the output gate.

Finally, the output vector is calculated by multiplying equations (6) and (7).

$$h_t = o_t \times tanh(C_t) \tag{7}$$

TABLE 2. Classification of the related works.

Related works	Method of learning approach	Machine learning models	Attacks simulated	Accuracy
[5]	Supervised learning	OCSVM	DoS, fuzzy, spoofing	>90%
[14]	Unsupervised learning	LSTM	plateau, playback, flooding, suppress, continuous change	>95%
[17]	Deep learning	GAN	DoS, fuzzy, RPM, gear	>95%
[19]	Deep learning	CNN	DoS, fuzzy, RPM, gear	>80%
[9]	Deep transfer learning	LeNet	Replay, flooding, fuzzy, spoofing	>98%
[16]	Transfer learning	DNN	Exploits, Fuzzers, DoS, Backdoor, Worms, Shellcode, etc.	>80%
[23]	Unsupervised learning	CNN + LSTM	Flood, replay, drop, spoof, fuzzy	>90%
[24]	Supervised learning	CNN + LSTM	DoS, spoofing gear, RPM, fuzzy	>99%
Proposed hybrid TL model	Transfer learning and fine-tuning	Hybrid CNN-LSTM TL model	DoS, fuzzy, RPM, gear, spoofing, flooding, replay	>99%

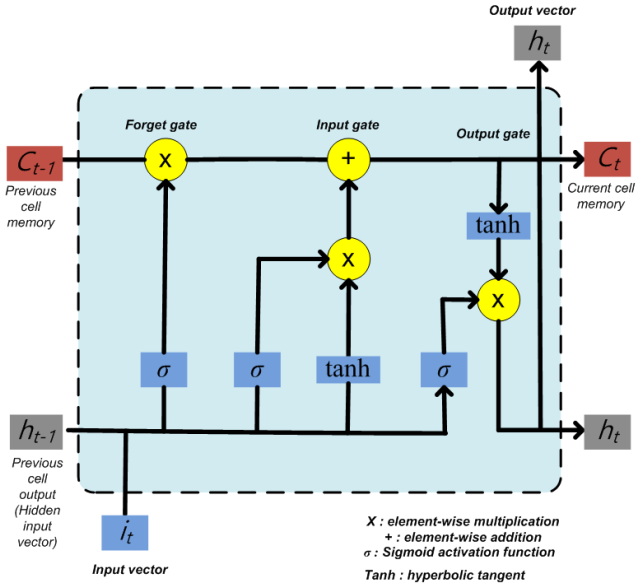

FIGURE 4. Long short-term memory (LSTM) network architecture.

B. PROBLEM STATEMENT

The vehicular network environment is unsafe and exposed to security vulnerabilities that must be appropriately addressed. Attackers can launch several attacks, such as flooding, spoofing, replay, and fuzzing attacks on CAN networks [9], [26].

- Flooding/DoS attack: In this attack, the hacker injects a large number of high-priority messages into the bus in a short period of time. This attack floods the bus with bogus messages, leading to the suppression of other valid message requests. This attack results from the ID-based priority scheme of the CAN network.
- Spoofing attack: In this form of attack, the hacker injects fake CAN IDs that appear similar to authentic CAN bus message IDs. It is launched to control specific vehicle system capabilities, such as brake control.
- Replay attack: Hackers gather CAN messages over a set time and replay them into the bus protocol.
- Fuzzing attack: Hackers inject bogus messages with faked IDs and random data values to disrupt the vehicle's normal operation.

Traditional machine learning algorithms and deep learning algorithms require a large number of training datasets for anomaly detection in CAN networks. On the other hand, the unavailability of large training datasets and computational complexity hinders the development of deep learning-based IDS systems for vehicular security. Furthermore, the IDS system should be dynamic and capable of detecting new attack types in a timely manner. The vulnerabilities in the CAN bus protocol can have serious consequences on the operation of fully autonomous vehicles governed by the V2X (vehicle-to-everything) technology in the near future. Thus, a different approach is needed to solve these issues. The following subsection provides a detailed explanation of the solution approach for mitigating the above-mentioned issues.

C. PROBLEM FORMULATION

Let us first define a domain and task. A domain D is comprised of a feature space x and a marginal probability distribution $P(X)$, where $X = \{x_1, x_2, x_3, \ldots\ldots, x_n\} \in x$ are the i^{th} vectors in the feature space [27].

Consider there is a domain D, which is given in equation (8).

$$D = \{x, P(X)\} \tag{8}$$

Given a D, a task is comprised of a label space y and an objective prediction function $f(\cdot)$, which is given in equation (9).

$$T = \{y, f(\cdot)\} \tag{9}$$

The task T is learned by the labeled training datasets. The training dataset consists of features and labels as $\{x_i, y_i\}$, where $x_i \in x$ and $y_i \in y$. The prediction function $f(x)$ can be used to predict the label for some feature vector x. Let D_s and D_t be the source and target domain, respectively. The features and labels in the source domain are represented as equation (10).

$$D_s = \{(x_s, y_s)\} = \{(x_{s_1}, y_{s_1}), (x_{s_2}, y_{s_2}), \ldots.., (x_{s_n}, y_{s_m})\} \tag{10}$$

where $x_{s_i} \in x_s$ are the features and $y_{s_i} \in y_s$ are the corresponding labels of those features. The labels can be either 0 (i.e., normal) or 1 (i.e., attack) for a binary classification problem.

Similarly, the features and labels in the target domain are represented as (11).

$$D_t = \{(x_t, y_t)\} = \{(x_{t_1}, y_{t_1}), (x_{t_2}, y_{t_2}), \ldots.., (x_{t_n}, y_{t_m})\} \tag{11}$$

where $x_{t_i} \in x_t$ is a feature and $y_{t_i} \in y_t$ is the corresponding label.

Consider a source dataset with a source domain D_s and learning task T_s, and a target dataset with a target domain D_t and learning task T_t. The goal of transfer learning is to maximize the objective function $f_t(\cdot)$ learning in the target domain D_t using the pre-trained features learned from D_s and T_s as expressed in equation (12) [27].

$$T_t = \{y_t, f_t(D_t|(D_s, T_s))\} \tag{12}$$

where $D_s \neq D_t$, or $T_s \neq T_t$.

The uniformity in the source and the target data samples is maintained by the Maximum Mean Discrepancy (MMD) equation, as expressed in (13) [9]. The equation calculates the difference between the source domain and the target domain. This strategy eliminates the need to train the machine learning model each time an attacker launches a new attack. The features learned from the source domain can be used to detect novel attacks in future IDS system development projects.

$$\text{Dist}(\mathbb{F}, x_s, x_t) := \sup_{f \in \mathbb{F}} \left(\frac{1}{n}\sum_{i=1}^{n} f(x_{s_i}) - \frac{1}{m}\sum_{i=1}^{m} f(x_{t_i})\right)^2 \tag{13}$$

where x_s and x_t represent the features, and notation n and m represent the number of features in the source dataset and the target dataset, respectively.

D. PROPOSED MODEL

This section proposes a deep hybrid transfer learning model for intrusion detection in CAN networks. Fig. 5 shows the proposed model. The figure shows that the proposed model consists of several stages: dataset preprocessing and feature extraction, machine learning model development and training, and anomaly detection. This paper further examines these stages in the following sections.

1) DATASET PREPROCESSING AND FEATURE EXTRACTION

Dataset preprocessing is the process of transforming raw data into a format that a computer can easily parse. This step is important for developing a machine-learning model and making accurate predictions. The unstructured real-world data comprises noises, null values, and redundant data. Using this data as direct input to an algorithm for feature interpretation is inappropriate. The use of noisy data for a prediction can lead to incorrect outcomes. Thus, dataset preparation is a crucial in maintaining data quality for data analysis. Various preprocessing techniques, such as standardization, handling of categorical variables, and one-hot encoding, have been used. The dataset used for this work was borrowed from the hacking and countermeasure laboratory research work and consists of the car hacking dataset [19] and car hacking: attack and defense challenge 2020 dataset [26]. The dataset samples were collected from a vehicle during stationary and driving mode for capturing patterns of CAN messages. The source and target datasets were produced for developing a transfer learning model. The source dataset consists of a large amount of data (approx. 70%), and the remaining 30% are used as the target dataset. The features used were Timestamp, Arbitration_ID, Data_Field, and Class (i.e., label) fields. First, one column data_field of eight bytes was split into eight-column data ranging from D[0] to D[7] with eight bits each. Duplicates were removed from the data, and the rows consisting of null values were dropped with the Python *dropna*() function. The hexadecimal values in the data were converted to base 16 integer form using the Python *int*() function.

The obtained CAN network dataset is in a CSV (Comma-Separated Values) format. On the other hand, the CNN model works better for the input of image data. Thus, we need to convert the low dimensional dataset into image form [28]. The CAN bus data is normalized to a scale of 0-255 because most image data pixel values are integers ranging between 0 and 255. There are various feature scaling methods in machine learning such as min-max normalization, standard scaler, robust scaler, L2 standardization, and quantile normalization. Lokman et al. investigated on the impact of different scaling methods on CAN network data [29]. Their analysis showed that quantile normalization is appropriate for

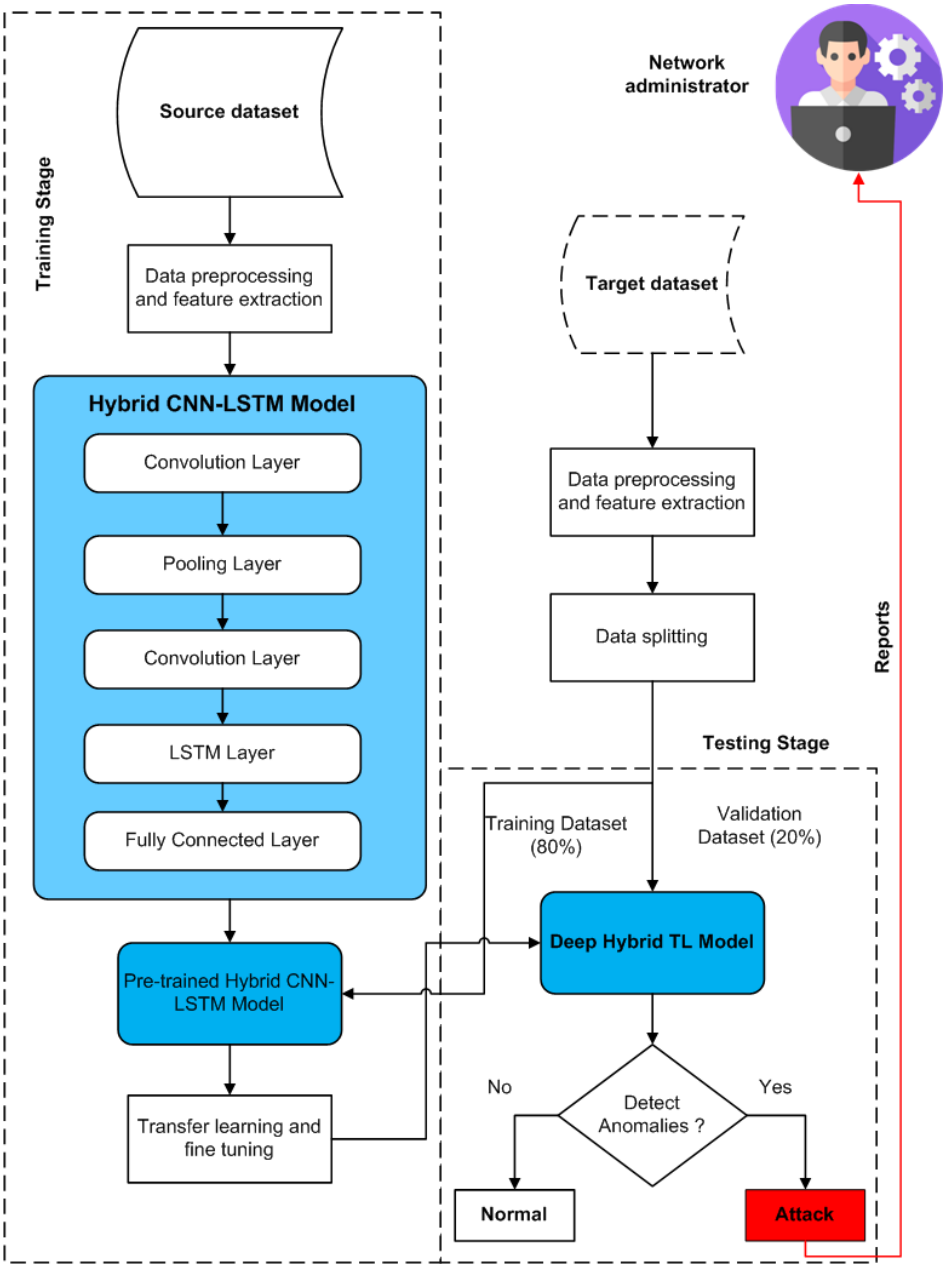

FIGURE 5. Proposed deep hybrid transfer learning model.

handling outliers compared to min-max and standardization techniques. Quantile normalization was used to scale the features in the proposed scheme. The quantile normalization is expressed in equation (14) [29].

$$inf\{x \in R : p \geq F(x)\} \tag{14}$$

Images were generated for each class of normal and intrusion patterns, including replay, flooding, DoS fuzzing, and spoofing attacks. The target dataset was divided into train and test datasets in the ratio of 80: 20 (i.e., 80% of the dataset is used for training, and the remainder is used for testing the TL model). The following subsections provide more details on the hybrid CNN-LSTM model development procedures.

2) HYBRID DEEP LEARNING MODEL DEVELOPMENT

A hybrid deep learning model based on CNN-LSTM was used to construct a pre-trained model. Fig. 5 depicts the use of a hybrid model in developing a transfer learning-based IDS system. Fig. 5 shows the complete procedure of the training and testing phases. The hybrid CNN-LSTM model is trained using the source dataset, which contains a large amount of CAN intrusion datasets. A hybrid model based on CNN-LSTM is used for this purpose. The hybrid model can learn CAN bus message features based on spatial and temporal feature extraction using the inbuilt capabilities of the CNN and LSTM models. The CAN bus message exhibits distinct patterns, and anomalous behaviors can be detected

by examining these properties. This study employed a source dataset that included DoS, flooding, replay, spoofing, and normal CAN message instances. Fuzzy attack datasets consist of IDs and message patterns that resemble the normal CAN message patterns. This confuses the IDS system and reduces the detection ability of the machine-learning models. Fuzzy attacks are difficult to detect because of this randomness in data patterns. Thus, the system aims to detect complex fuzzy attack types more accurately.

The hybrid model consists of convolution, pooling, LSTM, and fully connected dense layers connected with each other in sequence. The output from the first layer is passed as input to the next subsequent layer in the series until the output is generated. Fig. 6 shows the layerwise architecture of the proposed hybrid model. The total number of trainable parameters for the proposed model is 52,676. The input layer consists of pixel values with the shape of $224 \times 224 \times 3$. The input image has three color channels: red, green, and blue. The first ConvNet (convolutional neural network) calculates a two dimensional convolution given an input and four dimensional filter to produce an output shape of volume $222 \times 222 \times 32$ (width, height, and depth, respectively). The first convolutional layer (Conv2D) consists of 32 feature maps with a convolution filter size of 3×3. After the convolution operation, there is a pooling layer. The maxpooling operation is performed on the nonlinear feature maps obtained through the RELU activation function. The size of the maxpooling layer is 2×2 filters with a stride of two pixels. This operation will downsample each depth slice in the input by two pixels along the spatial domain (i.e., width and height), resulting in a volume of $111 \times 111 \times 32$. Adding the pooling layer, another convolution layer, produces a volume of $109 \times 109 \times 64$. The layers were downsampled to produce a volume of 64. The LSTM layer is added with the number of units as 64 and the activation function as RELU. Finally, the dense layer on top with four outputs is added for both CAN intrusion detection datasets. Table 3 lists the hyperparameters for the proposed hybrid CNN-LSTM model.

3) TRANSFER LEARNING AND FINE-TUNING

The TL IDS model is trained on a sufficiently large source dataset and can be used as the basic model for intrusion detection in target datasets containing unique attack types with a small number of training instances. With this method, the target model can use the learned feature maps from the source model instead of training from scratch. This strategy can also save training time and improve the performance of the IDS system. The general working of the TL-based IDS system is as follows.

- Utilize the layers from the pretrained model of the source dataset and freeze them to save the features learned for future training.
- Add new layers on top of the frozen layers that can be trained for the target dataset and learn the old features.

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 224, 224, 3)]	0
conv2d (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_1 (Conv2D)	(None, 109, 109, 64)	18496
dropout_2 (Dropout)	(None, 109, 109, 64)	0
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 64)	0
reshape_1 (Reshape)	(None, 1, 64)	0
lstm_1 (LSTM)	(None, 64)	33024
dropout_3 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 4)	260

Total params: 52,676
Trainable params: 52,676
Non-trainable params: 0

FIGURE 6. Architecture of the hybrid CNN-LSTM model.

- Fine tuning: unfreeze the top layers of the pre-trained model and train alongside the new classifier.

Freezing a model sets the weight of the model layer to non-trainable from trainable. This will prevent weights in the specified layer from being updated during training. This can be achieved with $pretrained\_model.trainable = False$ function in keras Dense layers. The approach to developing a TL model is a feature extraction and fine-tuning. In feature extraction, relevant features from the new samples are extracted using the images learned from the pre-trained network. A new classifier on top of the pretrained model was trained to reuse the feature mappings learned before for the dataset.

Fine-tuning is a part of the transfer learning process in which the pre-trained network weight (i.e., hybrid CNN-LSTM model) are used to train the new dataset. In this case, only the top layers of the hybrid CNN-LSTM model are used to train alongside the newly added classifier. In the event of fine-tuning, some top layers of the frozen base model are unfrozen, and train the newly added classifier layers. The top layers of the pre-trained model on the target dataset are trained simultaneously. The weight parameter is tuned in such a way that the proposed TL model will learn quality features for the new dataset. Fig. 7 shows the process of fine-tuning mechanism.

IV. EXPERIMENT SETUP AND PERFORMANCE EVALUATION

This section examines the experimental parameters and settings. The performance of the proposed TL-based IDS system is evaluated based on various machine-learning metrics. We used Python programming language and Keras deep learning API for machine learning model development. The experiment was performed on a Windows 11 Pro machine with an Intel Core i7-9750H CPU @ 2.60GHz processor, 16 GB RAM, and 1 TB hard drive.

TABLE 3. Hyperparameters used for the hybrid CNN-LSTM model development.

Parameters	Value
Convolution layer	Filters = 32, Kernel size = 3 × 3, activation function = RELU Filters = 64, Kernel size = 3 × 3, activation function = RELU, model = Sequential
Maxpooling layer	Filters = 2 × 2
CNN layer dropout	0.20
LSTM layer	Number of units = 128, activation function = softmax
LSTM layer dropout	0.25
Dense layer	activation function = softmax
no of epochs	10 (early stopping provided)
Learning rate	0.001
Loss function	categorical crossentropy
Optimizer	adam

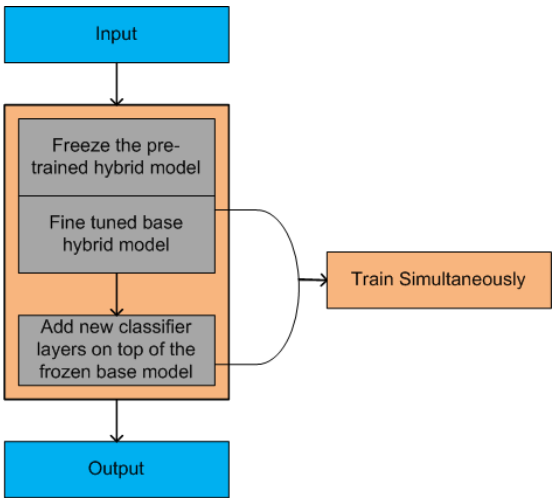

FIGURE 7. Procedure for transfer learning and fine-tuning.

A. EXPERIMENTAL DATASET

The intrusion detection system using a deep learning algorithm requires datasets containing the normal and intrusive behaviors of the CAN in-vehicle systems. Very few datasets are available online and are not usually public due to security concerns [15]. The hacking and countermeasure laboratory provides a car-hacking dataset that is open source and can be used for this purpose. This study used two sets of datasets provided by this community: i) the car-hacking dataset and ii) the car-hacking attack and defense challenge 2020 dataset for experimentation as listed in Table 4. Table 5 and 6 list the description of the dataset.

TABLE 4. Two datasets for our experiment.

Dataset	# Description
Type A	The car-hacking dataset
Type B	The car-hacking attack and defense challenge 2020 dataset

The attack types simulated in a car-hacking dataset are DoS, fuzzy, RPM spoofing, and gear spoofing attacks. In the DoS attack, high-priority message ID '0000' is injected in 0.3-millisecond intervals. In a fuzzy attack, random CAN ID

TABLE 5. The car-hacking dataset description.

Attack type	# Of normal messages	# Of attack messages
DoS Attack	3,078,250	587,521
Fuzzy Attack	3,347,013	491,847
RPM Spoofing	3,966,805	654,897
Gear Spoofing	3,845,890	597,252
Attack-free (normal)	988,872	-

and data values are injected every 0.5 milliseconds. Every 1-millisecond, spoofed CAN IDs and messages are injected in the spoofing attack dataset. These datasets constitute 30-40 minutes of CAN messages. The attack types simulated in a car hacking attack and defense challenge 2020 dataset are flooding, spoofing, replay, and fuzzing attacks. This study used the preliminary round training dataset containing normal and four types of attacks consisting of 3,672,151 instances, among which 299,408 instances were the attack types for car hacking: attack and defense challenge 2020. The authors launched these attacks on a real vehicle (Hyundai Avante CN7 for the car hacking: attack & defense challenge 2020 dataset) by connecting Raspberry Pi devices to the CAN networks to inject fabricated messages via the OBD-II port. Each dataset is in CSV format and consists of attributes such as timestamp, arbitration ID, data length code (DLC), data (D[0] - D[7], each with eight bytes), and class.

The efficacy of the proposed transfer learning model was evaluated by training the proposed model on a source dataset with only normal data and three attack types: flooding, spoofing, and replay attacks for car hacking attack and defense challenge 2020 dataset. The source model in the case of the car hacking dataset was developed based on a normal dataset and three attack types: DoS, spoofing RPM, and spoofing gear attack dataset. The fuzzy attack was kept silent in source model training and is used to assess the performance of transfer learning models. The fuzzy attack dataset contains arbitrary IDs and data that a machine learning algorithm would struggle to learn. The main goal of this paper is to develop a transfer learning-based IDS system, which can learn the features from the pre-trained hybrid CNN-LSTM

TABLE 6. The car-hacking: attack and defense challenge 2020 dataset description.

Description	# Of normal messages	# Of attack messages
Normal and four types of attacks (flooding, spoofing, replay, and fuzzing) labelled dataset	3,372,743	299,408

model and use it to detect new attack types with higher detection accuracy.

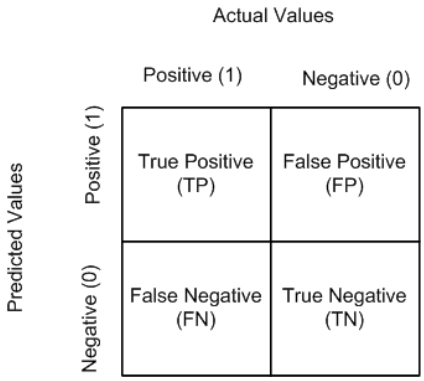

FIGURE 8. Confusion matrix.

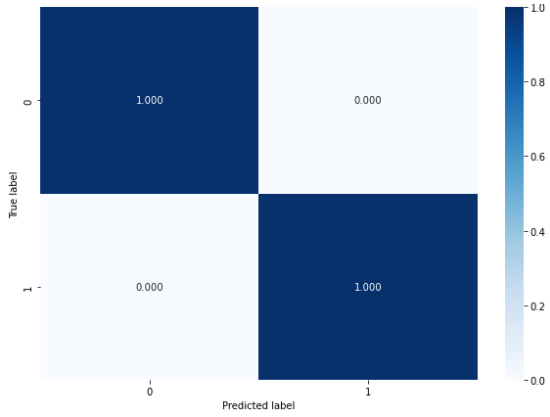

FIGURE 9. Confusion matrix of the proposed hybrid TL model for fuzzy attack detection (Type A dataset).

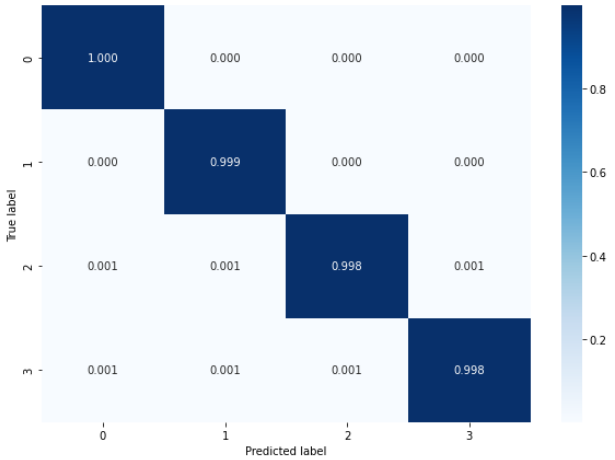

FIGURE 10. Confusion matrix of the proposed hybrid TL model for varying attack detection (Type A dataset).

B. EVALUATION METRICS

The evaluation metrics for the proposed IDS system are derived from the confusion matrix or an error matrix. This matrix structure will allow visualization of the performance of the classification algorithm. The classification matrix output can be binary and multiclass based on the dataset and the algorithm used for prediction. The confusion matrix provides various indicators such as true positive (TP), true negative (TN), false positive (FP), and false negative (FN), for measuring the performance, as shown in Fig. 8. TP is the test results of the model predictions where it correctly predicts the positive class. Similarly, TN is defined as the correct prediction for the negative class. FP represents incorrect predictions for the positive class. FN comprises the model predictions that provide incorrect predictions for the negative class. The confusion matrix indicators provide the evaluation metrics for the IDS interms of accuracy, precision, recall, and f1-score. The ratio of accurately anticipated observations to the total observations is defined as accuracy. The ratio is calculated as (15). Precision is the ratio of true positives over the total number of positive predictions by the machine learning model, calculated using equation (16). Recall or sensitivity measures true predictions over the number of actual positive outcomes. These measures are computed using equation (17). The F1-score is the harmonic mean between precision and recall, which can be calculated using equation (18).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

$$Recall/Sensitivity = \frac{TP}{TP + FN} \tag{17}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{18}$$

C. RESULTS AND DISCUSSION

This subsection reports the results and explanation of the suggested TL-based IDS model. The accuracy, precision, recall, f1-score, false positive rate (FPR), and ROC curves are used to assess the model efficacy. The proposed model

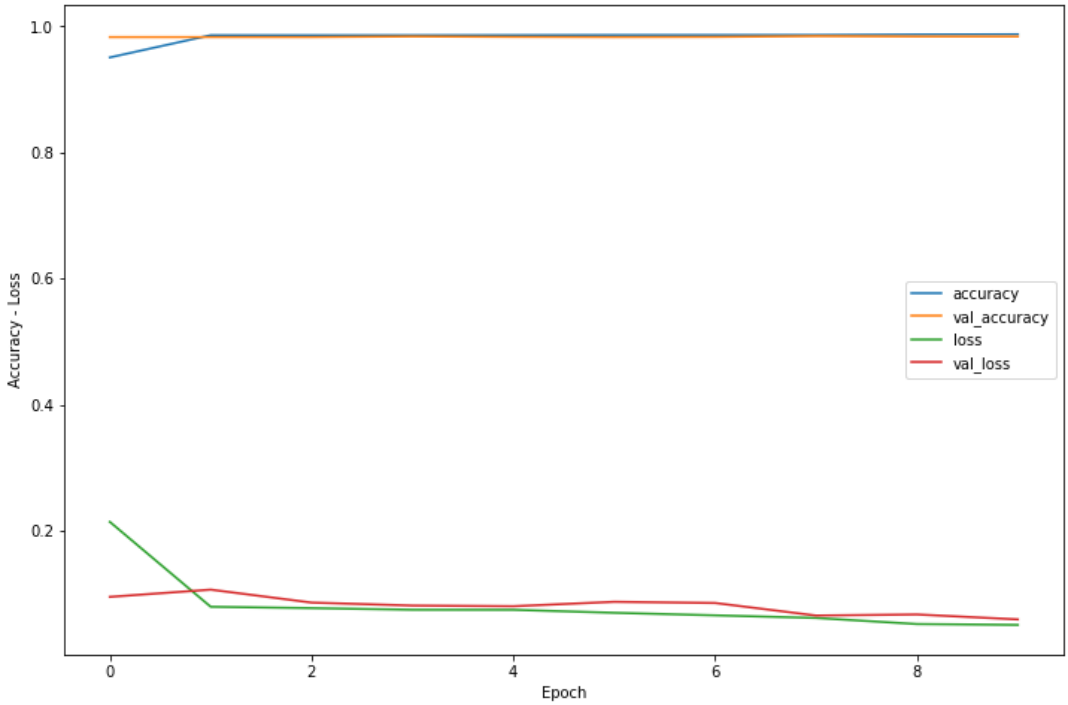

FIGURE 11. Training and validation accuracy/loss vs. epochs of the proposed hybrid TL model (Type A dataset).

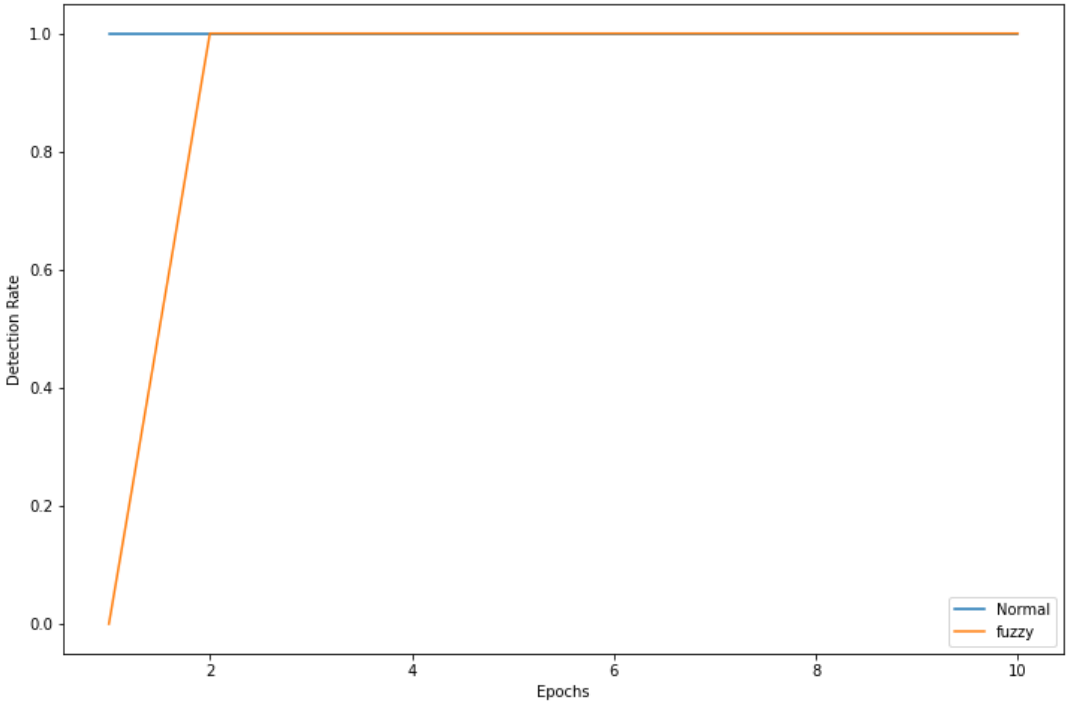

FIGURE 12. Detection rate of the proposed hybrid TL model for fuzzy attack dataset (Type A dataset).

was compared to cutting-edge IDS systems for in-vehicle network intrusion detection. The proposed hybrid TL model was compared with the state-of-the-art algorithms [9], [17], and [19]. Song et al. developed an IDS system based on a CNN deep learning algorithm [19]. They used the inception-ResNet model for intrusion detection for varying datasets simulating normal and attack scenarios. Seo et al. developed the IDS system using generative adversarial networks [17]. In contrast, Mehdi et al. employed the LeNet model for intrusion detection [9]. The results of the

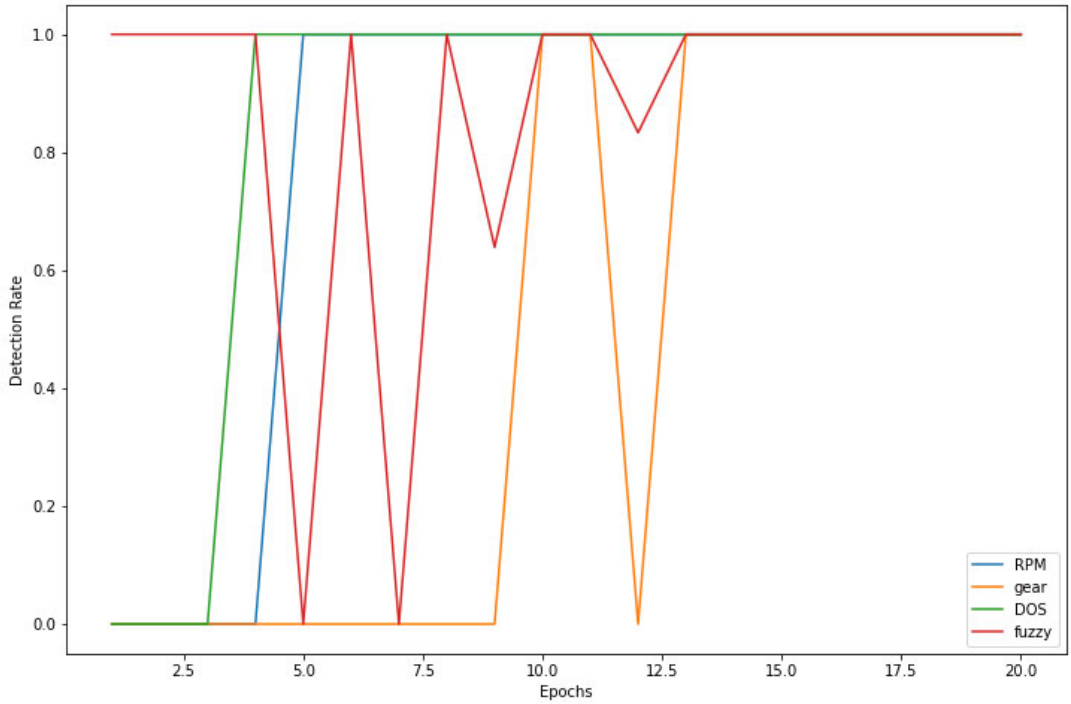

FIGURE 13. Detection rate vs. Epochs of the proposed hybrid TL model for varying attack types (Type A dataset).

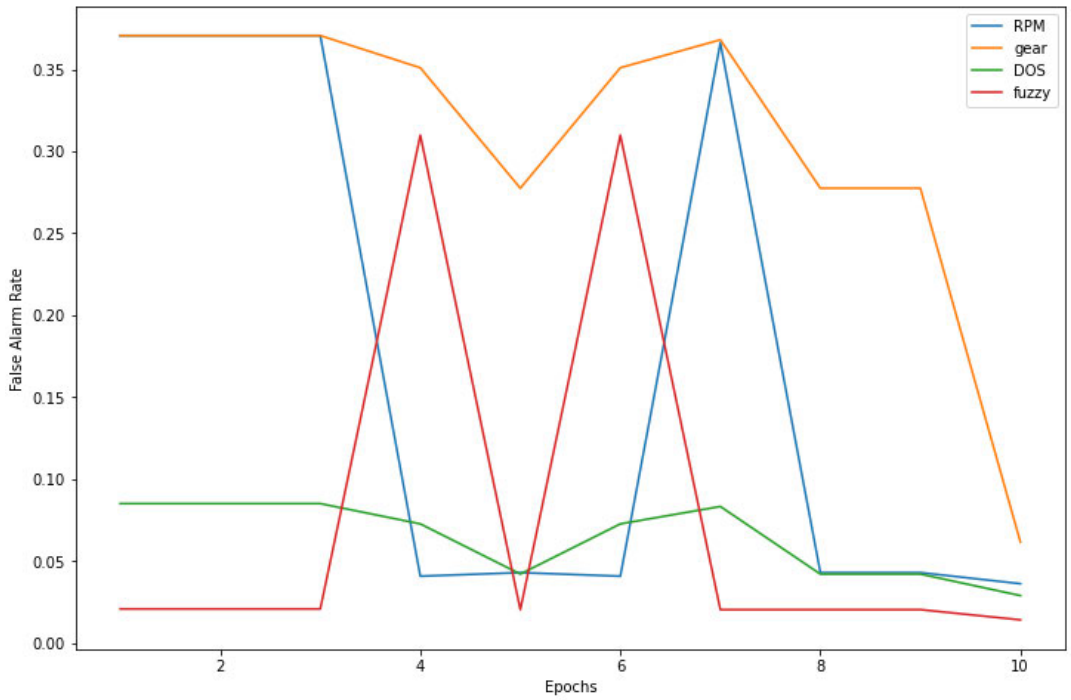

FIGURE 14. False alarm rate vs Epochs of the proposed hybrid TL model for varying attack types (Type A dataset).

proposed work are also compared to the performance of CNN and LSTM-based IDS systems. The following subsections present the experimental findings of the proposed model on two benchmark datasets: car-hacking and car-hacking: attack & defense challenge 2020.

1) RESULTS FOR CAR HACKING DATASET

This subsection provides the results of the proposed TL model for the car hacking dataset (i.e., Type A dataset). Fig. 9 shows the confusion matrix of the proposed model for fuzzy attack detection. The normal and fuzzy attack

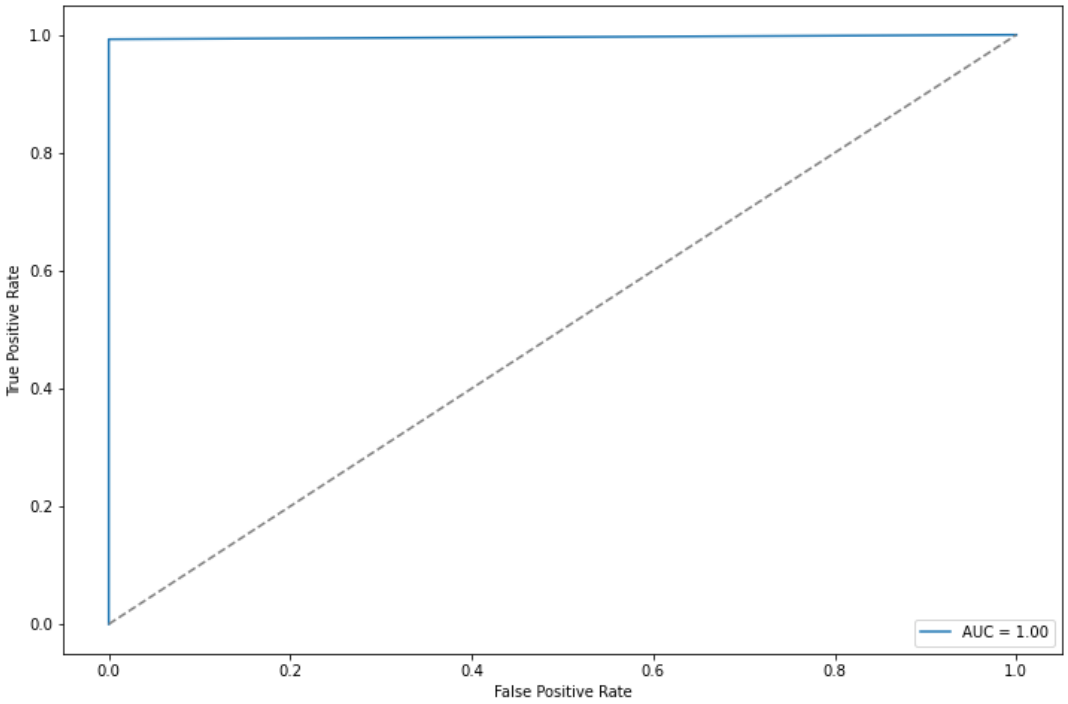

FIGURE 15. ROC curve of the proposed hybrid TL model for fuzzy attack detection (Type A dataset).

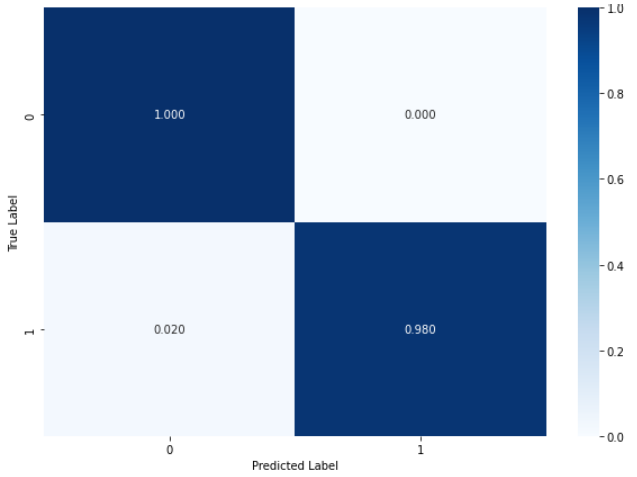

FIGURE 16. Confusion matrix of the proposed hybrid TL model on fuzzy attack detection (Type B dataset).

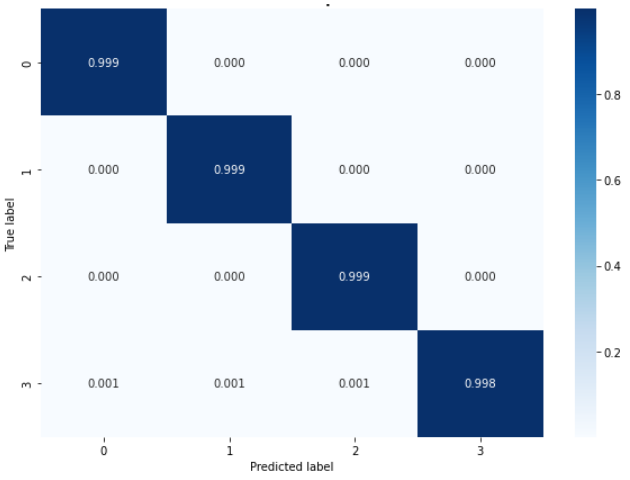

FIGURE 17. Confusion matrix of the proposed hybrid TL model for varying attack detection (Type B dataset).

types have been labeled as 0 and 1, respectively, in Fig. 9. Results shows that the detection accuracy of the model for fuzzy attack is 100%. The confusion matrix for varying types of attacks detection is shown in Fig. 10. The labels 0, 1, 2, and 3 in the confusion matrix of Fig. 10 refers to rpm, gear, DoS and fuzzy attack patterns, respectively. The confusion matrix plot shows that the proposed model can achieve better performance of 99.9% accuracy for varying attack types. Fig. 11 shows the training and validation accuracy/loss for diverse epochs for the proposed hybrid TL model on the car hacking dataset. The accuracy loss plot shows that the model can achieve convergence for the accuracy and validation accuracy plots and the loss and validation loss plots at approximately 10 epochs. Figs. 12 and 13 show the detection rate of the proposed hybrid TL model for single-attack detection and multiple-attack detection. The false alarm rate vs. the number of epochs is given in Fig. 14. Fig. 15 shows the ROC curve of the proposed model. The ROC AUC curve value is 1.0. The results showed that the false positive rate of the proposed model is significantly lower for all sets of the datasets.

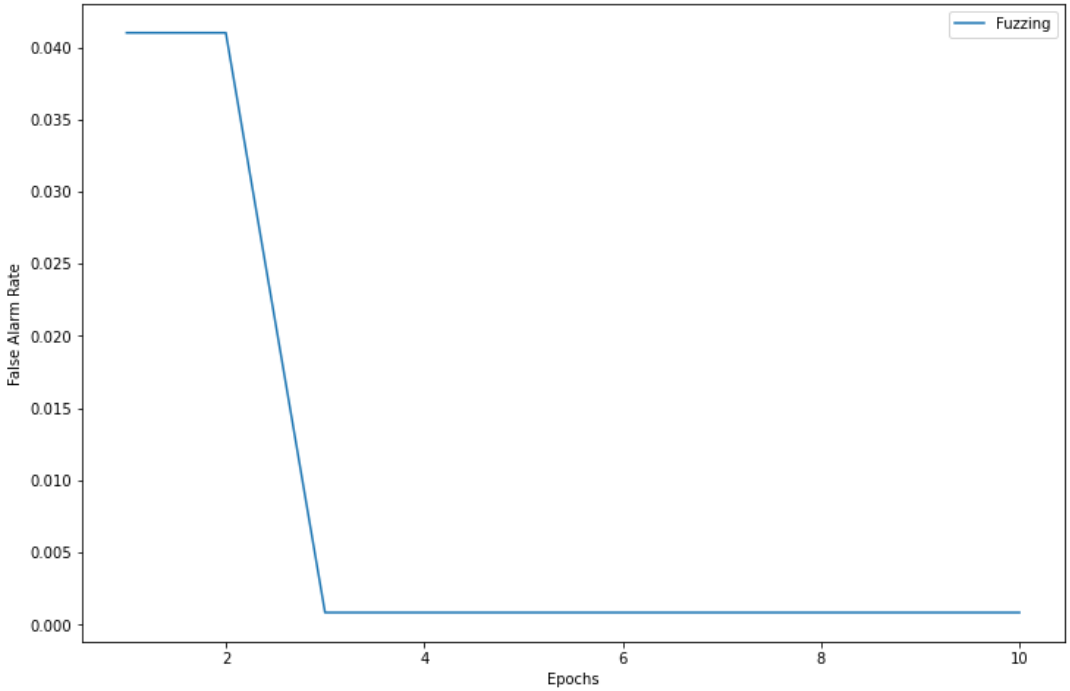

FIGURE 18. False alarm rate of the proposed hybrid TL model for the fuzzy attack detection (Type B dataset).

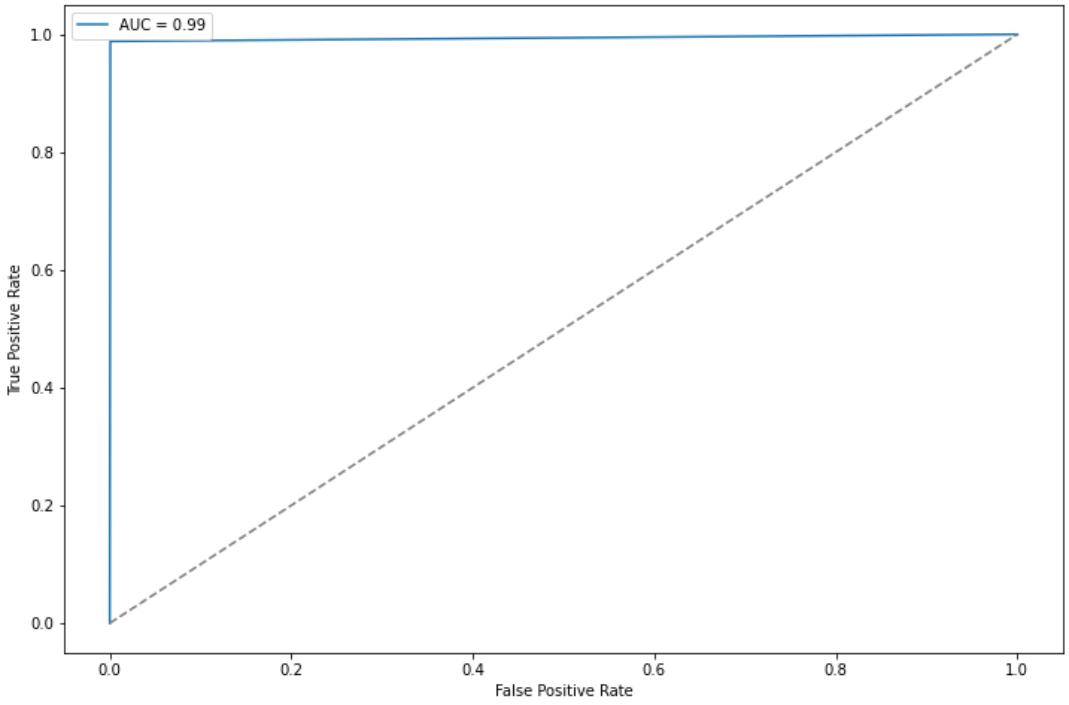

FIGURE 19. ROC curve of the proposed hybrid TL model for the fuzzy attack detection (Type B dataset).

2) RESULTS FOR CAR HACKING: ATTACK AND DEFENSE CHALLENGE 2020 DATASET

This subsection provides the experimentation results of the proposed model performed on the car hacking: attack & defense challenge 2020 dataset (i.e., Type B dataset). Fig. 16 provides the confusion matrix for fuzzy attack detection. Fig. 17 shows the confusion matrix of the proposed model on the presence of varying attack patterns. The labels 0, 1, 2, and 3 in the confusion matrix of Fig. 17 refer to replay, flooding, spoofing, and fuzzing attack patterns, respectively.

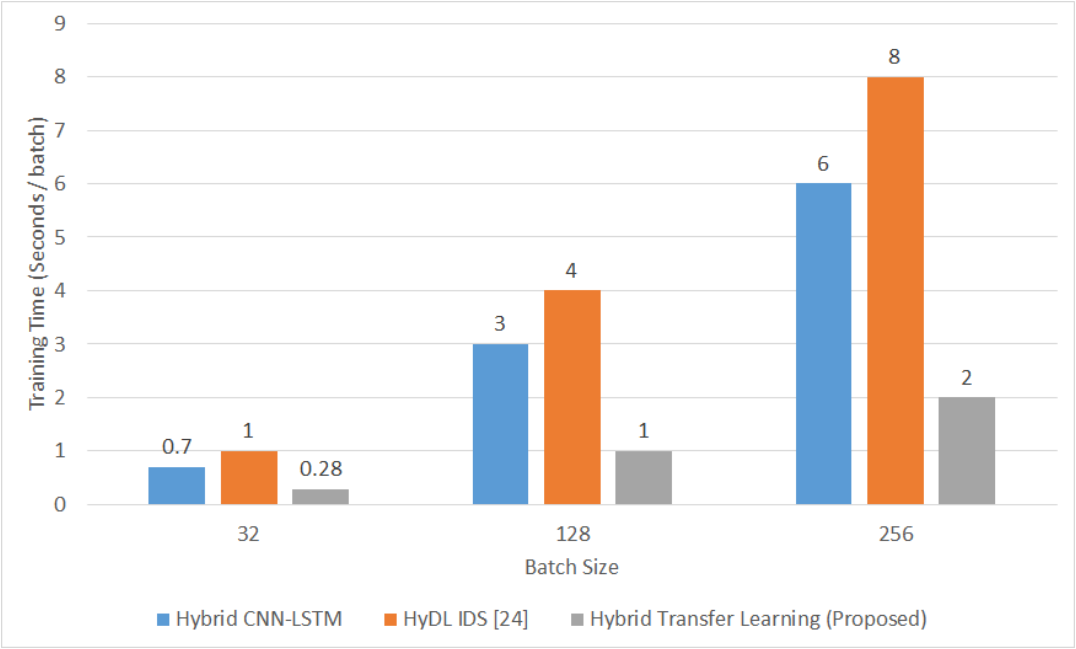

FIGURE 20. Training time comparison results (Type A dataset).

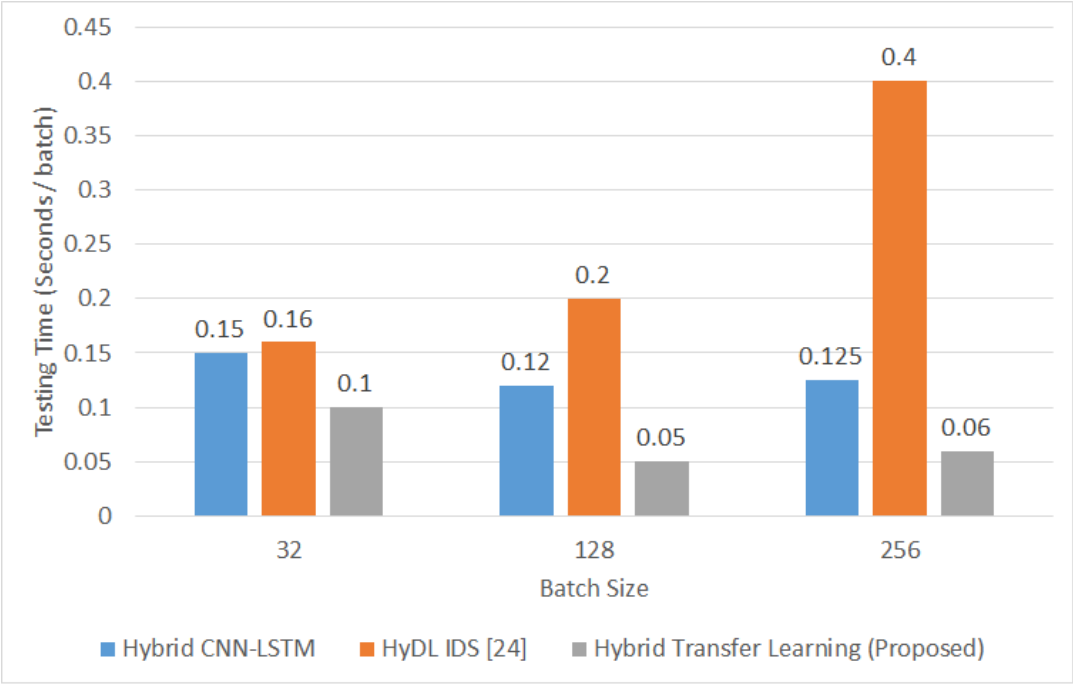

FIGURE 21. Testing time comparison results (Type A dataset).

The confusion matrix result shows that the model have 99.9% accuracy, precision, recall, and f1-scores. Fig. 18 shows the false alarm rate of the proposed model is in the range of 0.00 on Type B dataset. Finally, the ROC AUC curve is shown in Fig. 19. The ROC AUC value is 0.99 which shows the model has higher prediction scores. These experimental results show that the proposed model has high detection accuracy and low false alarm rates based on the comparative analysis with the related works in the corresponding domain.

Tables 7 and 8 provide the comparison results of the proposed work with related works [9], [17], [19], [24] in CAN in-vehicle security. The results show that the proposed model achieved higher accuracy (100%), precision(100%), recall(100%), f1-score(100%), ROC AUC(1.0), and lower FPR(0.0) for fuzzy attack detection (Type A dataset),

TABLE 7. Results comparison with the state-of-the-art algorithms for fuzzy attack detection for car hacking dataset.

Reference	Accuracy	precision	recall	f1-score	FPR	ROC AUC
Song et al. [19]	-	0.9995	0.9965	0.9980	-	-
Seo et al. [17]	0.9800	0.9730	-	-	-	0.9990
Lo et al. [24]	0.9998	0.9998	0.9988	0.9993	0.0021	0.9900
Proposed Hybrid TL Model	1.0000	1.0000	1.0000	1.0000	0.0000	1.0000

TABLE 8. Results comparison with the state-of-the-art algorithms for fuzzy attack detection for car hacking: attack & defense challenge 2020 dataset.

Reference	Accuracy	precision	recall	f1-score	FPR	ROC AUC
Mehdi et al. [9]	0.9810	0.9814	0.9804	0.9783	-	0.9542
Proposed Hybrid TL Model	0.9991	0.9991	0.9991	0.9991	0.0000	0.9900

TABLE 9. Comparative results of the proposed hybrid TL model with various deep learning algorithms for car hacking dataset.

Algorithm	Accuracy	precision	recall	f1-score	FPR	ROC AUC
CNN	0.9977	0.9977	0.9804	0.9941	0.0023	0.9500
LSTM	0.9977	0.9977	0.9997	0.9997	0.0023	0.9500
Lo et al. [24]	0.9996	0.9996	0.9996	0.9996	0.0023	0.9900
Proposed Hybrid TL Model	0.9997	0.9997	0.9997	0.9997	0.0001	0.9900

TABLE 10. Comparative results of the proposed hybrid TL model with various deep learning algorithms for car hacking: attack & defense challenge 2020 dataset.

Algorithm	Accuracy	precision	recall	f1-score	FPR	ROC AUC
CNN	0.9664	0.9340	0.9664	0.9499	0.0336	0.8000
LSTM	0.9645	0.9304	0.9645	0.9471	0.0355	0.8200
Lo et al. [24]	0.9965	0.9968	0.9965	0.9965	0.0500	0.9700
Proposed Hybrid TL Model	0.9991	0.9991	0.9991	0.9991	0.0005	0.9900

compared to state-of-the-art algorithms. These results confirmed the efficacy of the transfer learning approach and its ability to learn the features from the previously trained model. This transfer of knowledge from one set of datasets to another datasets with new instances of attack patterns can be an evolutionary approach for vehicular network intrusion detection systems. Tables 9 and 10 compare the proposed algorithm with CNN, LSTM, and HyDL IDS [24] algorithms based on multiclass classification using both datasets. Good results were achieved by our model using transfer learning and tuning the appropriate weight during the fine-tuning procedure. In addition, suitable temporal and space-related characteristics were retrieved and learned from the pre-trained hybrid CNN-LSTM model, which aided better classification. The overall accuracy of the proposed model for attack detection is 99.9% according to experimentation. Whereas the use of individual CNN, LSTM, and hybrid HyDL IDS resulted in lower accuracy results.Thus, the results show that the proposed algorithm has advantages over the other algorithms. Results prove that the proposed work is efficient for fuzzy attack detection and other sets of attack-type detection, such as DoS, spoofing, replay, rpm, and gear. These results show that the proposed TL model is dynamic and can detect various attack types with higher accuracy.

The intrusion detection system should be dynamic and be able to detect new attack types with a higher detection accuracy and lower false positive rates. The proposed hybrid TL model-based IDS system can attain these characteristics. The experiment results show that the proposed system can detect more than one attack type with higher accuracy. Furthermore, the false positive rate of the proposed model is significantly lower than the previous works. Thus, this system is considered as a robust IDS system. This has been proved through experimentation and the generated results.

3) ANALYSIS OF TRAINING AND TEST TIME

This section compares the training and testing times for the hybrid model utilizing the proposed transfer-learning (TL)

approach to those for the hybrid CNN-LSTM and the HyDL IDS [24]. Fig. 20 and Fig. 21 compare the transfer learning strategy's training and testing time to the hybrid CNN-LSTM approach based on batch size. The results suggest that by deploying a hybrid transfer learning-based IDS system, training and testing costs can be reduced. Training time for the hybrid CNN-LSTM model and HyDL IDS increased significantly as batch size increased. However, as batch size grows, the training and testing time of the Hybrid TL model is lowered by more than 30%. As demonstrated in Fig. 21, the hybrid TL model has the shortest testing duration compared to other approaches. As a result, the suggested IDS system is effective and can reduce training and testing time when compared to simple CNN-LSTM-based IDS systems. The analysis of training and testing time have been performed based on Type A dataset.

V. CONCLUSION AND FUTURE WORKS

The vehicle network environment is not safe in today's connected world. With the developments in communication technologies and the need to provide convenient services to passengers, there is an enormous rise in the security risk to a vehicle. On the other hand, manufacturers do not prioritize the security of vehicles. There is an obvious trend of an increase in the number of smart and autonomous vehicles. Although manufacturers are developing smart and autonomous vehicles that can perform multiple functions efficiently, the security risks associated with these technologies have not been thoroughly investigated. This paper offers a unique intrusion detection approach based on transfer learning. The proposed model has higher accuracy, precision, recall, and f1-score (nearly 99.9% for each dataset), according to the results. Furthermore, the TL technique reduces training and testing time by more than 30% when compared to state-of-the-art algorithms. The hybrid TL model can overcome the weaknesses of present IDS systems and be a candidate for future IDS systems because of its potential to detect novel attack types with improved detection capabilities. The experimental findings on two sets of datasets revealed the superiority of the proposed methodology over state-of-the-art methods. The results have been evaluated based on various machine learning metrics, such as accuracy, precision, recall, f1-score, false positive rate, and area under the curve (AUC).

In the future, we will study the feasibility of transfer learning approach in CAN networks under the influence of malicious nodes launching sophisticated attack patterns.

REFERENCES

[1] N. Khatri, R. Shrestha, and S. Y. Nam, "Security issues with in-vehicle networks, and enhanced countermeasures based on blockchain," *Electronics*, vol. 10, no. 8, p. 893, Apr. 2021.

[2] T. Hoppe, S. Kiltz, and J. Dittmann, "Security threats to automotive can networks—Practical examples and selected short-term countermeasures," *Rel. Eng. Syst. Saf.*, vol. 96, no. 1, pp. 11–25, 2011.

[3] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, vol. 2015, no. S 91, pp. 1–91, Aug. 2015.

[4] S. Nie, L. Liu, Y. Du, and W. Zhang, "Over-the-air: How we remotely compromised the gateway, BCM, and autopilot ECUS of Tesla cars," in *Proc. Briefing, Black Hat USA*, 2018, pp. 1–19.

[5] O. Avatefipour, A. S. Al-Sumaiti, A. M. El-Sherbeeny, E. M. Awwad, M. A. Elmeligy, M. A. Mohamed, and H. Malik, "An intelligent secured framework for cyberattack detection in electric vehicles' CAN bus using machine learning," *IEEE Access*, vol. 7, pp. 127580–127592, 2019.

[6] M. D. Hossain, H. Inoue, H. Ochiai, D. Fall, and Y. Kadobayashi, "LSTM-based intrusion detection system for in-vehicle can bus communications," *IEEE Access*, vol. 8, pp. 185489–185502, 2020.

[7] N. Khatri, S. Lee, A. Mateen, and S. Y. Nam, "Event message clustering algorithm for selection of majority message in VANETs," *IEEE Access*, vol. 11, pp. 14621–14635, 2023.

[8] R. Shrestha, S. Y. Nam, R. Bajracharya, and S. Kim, "Evolution of V2X communication and integration of blockchain for security enhancements," *Electronics*, vol. 9, no. 9, p. 1338, Aug. 2020.

[9] S. T. Mehedi, A. Anwar, Z. Rahman, and K. Ahmed, "Deep transfer learning based intrusion detection system for electric vehicular networks," *Sensors*, vol. 21, no. 14, p. 4736, Jul. 2021.

[10] R. Shrestha, R. Bajracharya, A. P. Shrestha, and S. Y. Nam, "A new type of blockchain for secure message exchange in VANET," *Digit. Commun. Netw.*, vol. 6, no. 2, pp. 177–186, May 2020.

[11] R. Shrestha and S. Y. Nam, "Regional blockchain for vehicular networks to prevent 51% attacks," *IEEE Access*, vol. 7, pp. 95033–95045, 2019.

[12] R. Shrestha and S. Y. Nam, "Trustworthy event-information dissemination in vehicular ad hoc networks," *Mobile Inf. Syst.*, vol. 2017, pp. 1–16, Mar. 2017.

[13] N. Sangeeta and S. Y. Nam, "Blockchain and interplanetary file system (IPFS)-based data storage system for vehicular networks with keyword search capability," *Electronics*, vol. 12, no. 7, p. 1545, Mar. 2023.

[14] M. Hanselmann, T. Strauss, K. Dormann, and H. Ulmer, "CANet: An unsupervised intrusion detection system for high dimensional CAN bus data," *IEEE Access*, vol. 8, pp. 58194–58205, 2020.

[15] A. Gazdag, S. Lestyán, M. Remeli, G. Ács, T. Holczer, and G. Biczók, "Privacy pitfalls of releasing in-vehicle network data," *Veh. Commun.*, vol. 39, Feb. 2023, Art. no. 100565.

[16] A. Singla, E. Bertino, and D. Verma, "Overcoming the lack of labeled data: Training intrusion detection models using transfer learning," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, Jun. 2019, pp. 69–74.

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

[18] Y. Lin, C. Chen, F. Xiao, O. Avatefipour, K. Alsubhi, and A. Yunianta, "An evolutionary deep learning anomaly detection framework for in-vehicle networks–CAN bus," *IEEE Trans. Ind. Appl.*, early access, 2020, Jul. 17, 2020, doi: 10.1109/TIA.2020.3009906.

[19] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Veh. Commun.*, vol. 21, Jan. 2020, Art. no. 100198.

[20] 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.*, Mar. 2020, pp. 1048–1055.

[21] Y. Otoum, Y. Wan, and A. Nayak, "Transfer learning-driven intrusion detection for Internet of Vehicles (IoV)," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, 2022, pp. 342–347.

[22] Z. Khademi, F. Ebrahimi, and H. M. Kordy, "A transfer learning-based CNN and LSTM hybrid deep learning model to classify motor imagery EEG signals," *Comput. Biol. Med.*, vol. 143, Apr. 2022, Art. no. 105288.

[23] H. Sun, M. Chen, J. Weng, Z. Liu, and G. Geng, "Anomaly detection for in-vehicle network using CNN-LSTM with attention mechanism," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 10880–10893, Oct. 2021.

[24] W. Lo, H. Alqahtani, K. Thakur, A. Almadhor, S. Chander, and G. Kumar, "A hybrid deep learning based intrusion detection system using spatial–temporal representation of in-vehicle network traffic," *Veh. Commun.*, vol. 35, Jun. 2022, Art. no. 100471.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[26] H. Kang, B. I. Kwak, Y. H. Lee, H. Lee, H. Lee, and H. K. Kim, "Car hacking and defense competition on in-vehicle network," in *Proc. 3rd Int. Workshop Automot. Auto. Vehicle Secur.*, 2021, p. 25.
[27] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2009.
[28] F. Hussain, S. G. Abbas, M. Husnain, U. U. Fayyaz, F. Shahzad, and G. A. Shah, "IoT DoS and DDoS attack detection using ResNet," in *Proc. IEEE 23rd Int. Multitopic Conf. (INMIC)*, Nov. 2020, pp. 1–6.
[29] S.-F. Lokman, A. T. Othman, M. H. A. Bakar, and S. Musa, "The impact of different feature scaling methods on intrusion detection for in-vehicle controller area network (CAN)," in *Advances in Cyber Security*. Penang, Malaysia: Springer, 2020, pp. 195–205.

NARAYAN KHATRI received the B.E. degree from Purbanchal University, Nepal, in 2013, and the M.S. degree in computer engineering from Yeungnam University, South Korea, in 2017, where he is currently pursuing the Ph.D. degree in information and communication engineering. He is a Research Assistant with the Computer Network and Security Laboratory, Department of Information and Communication Engineering. His research interests include network security, blockchain, vehicular ad hoc networks, and machine learning for network intrusion detection systems.

SIHYUNG LEE received the B.S. (summa cum laude) and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 2000 and 2004, respectively, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University (CMU), USA, in 2010. From 2010 to 2011, he was a Postdoctoral Researcher with the Network Management Group, IBM Thomas J. Watson Research Center, USA. From 2011 to 2019, he was a Professor with the Department of Information Security, Seoul Women's University, South Korea. Since 2019, he has been a Professor with the School of Computer Science and Engineering, Kyungpook National University. His research interests include pattern mining from social network traffic and program synthesis for classical and quantum computers.

SEUNG YEOB NAM (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 1997, 1999, and 2004, respectively. From 2004 to 2006, he was a Postdoctoral Research Fellow with the CyLab, Carnegie Mellon University. In 2007, he joined the Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, South Korea, where he is currently a Professor. From January 2012 to January 2013, he was a Visiting Professor with the Department of Electrical and Computer Engineering, Carnegie Mellon University. His research interests include network security, blockchain, network management, and wireless networks. He received the 2022 Best Paper Award from Digital Communications and Networks (DCN).