A Transfer Learning Based Intrusion Detection System for Internet of Vehicles

Achref Haddaji, Samiha Ayed, Lamia Chaari Fourati

A Transfer Learning Based Intrusion Detection System for Internet of Vehicles

Achref Haddaji
LIST3N-ERA, University of Technology of Troyes, France
National School of Electronics and Telecommunications of Sfax, Sfax, Tunisia
Digital Research Center of Sfax (CRNS), Laboratory of Signals, systeMs, aRtificial Intelligence, neTworkS (SM@RTS), Sfax University; TUNISIA
Email: achref.haddaji@utt.fr

Samiha Ayed
LIST3N-ERA, University of Technology of Troyes, France
Email: samiha.ayed@utt.fr

Lamia Chaari Fourati
Digital Research Center of Sfax (CRNS), Laboratory of Signals, systeMs, aRtificial Intelligence, neTworkS (SM@RTS), Sfax University; TUNISIA
Email:lamiachaari1@gmail.com

***Abstract*—With the fast expansion of the internet of vehicles (IoV) and the emergence of new types of threats, the traditional machine learning-based intrusion detection systems must be updated to meet the security requirements of the current environment. Recently, deep learning has shown exceptional performance in IoV intrusion detection. However, deep learning-based intrusion detection system (DL-IDS) models are more fixated and dependent on the training dataset. In addition, the behavior changes with the occurrence of attacks. They pose a real problem for the DL-IDS and make their detection more complicate. In this paper, we present a deep transfer learning based intrusion detection in-vehicle (TRLID) model for IoV using the CAN bus protocol. In our proposed model, a data preparation approach is proposed to clean up bus data and convert it to an image for usage as input to the deep learning model. Indeed, we used transfer learning characteristics because they enable us to transfer the source task's knowledge to the target task. Therefore, we trained our model using different dataset including different attacks. The experimental results show that our proposed TRLID achieved good results where the intelligence integration of transfer learning was efficient for attacks detection.**

IoV security, Transfer Learning, CAN Bus attacks, Intrusion Detection

I. Introduction

Recently, the intelligent transportation systems (ITS) in many countries are being gradually expanded to their maximum potential as the number of users continue to increase. The primary goal of these transportation systems are to increase traffic monitoring, road safety and passenger comfort in order to reduce accidents [1]. VANET, or vehicular ad hoc networks [2], have been designed as the first ITS to keep the driver informed about real time traffic through exchanging information messages. In VANET two types of communication are possible: vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I). The main purpose of VANET is to improve traffic efficiency by decreasing communication time, cost and pollution emissions. However, there are still a number of VANET related emerging issues that need to be solved in existing vehicle networks. Some of these problems are inconsistent internet access, compatibility issues with personal devices, processing power limitations and the lack of cloud computing services, among others. To address these issues, Internet of Vehicles (IoV) [3] which is an emerging system in the ITS were proposed via combining VANET with Internet of things (IoT). The adoption of IoV can effectively enhance smart cities and sustainable energy growth. For instance, the traffic management system in the IoV and autonomous driving vehicle environment could reduce traffic congestion, traffic accidents, and industrial contamination. Intelligent transportation system-networked vehicles can also provide users with a more comprehensive and personalized mobile transportation service. These services include route planning, service recommendation, and intelligent parking, thereby enhancing the transportation efficiency. Mainly, they

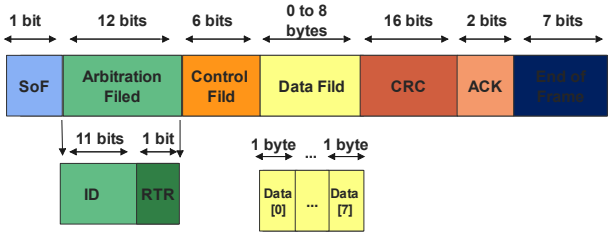


Fig. 1: Structure of CAN frame

are figured out as a heterogeneous vehicular environment with different communication types such as V2V, V2I, vehicle-to-pedestrian (V2P), vehicle-to-sensors (V2S), and vehicle-to-network (V2N). Moreover, there are various wireless technologies [4] used by IoVs in order to establish an effective communication, including vehicular communications such as DSRC/CALM, cellular mobile communication such as 4G, LTE, WiMax, and Satellite, and short-range static communication such as Bluetooth and Wi-Fi. The based systems of IoVs mainly compromise the external networks and the

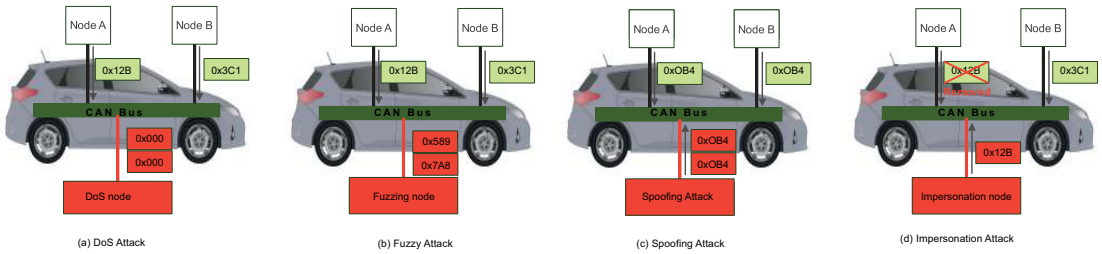


Fig. 2: Four attack scenarios on a CAN bus

intra-vehicle networks (IVNs). Indeed, the external networks enable the connection between smart cars and the other entities, such as the infrastructure. Meanwhile, the intra-vehicle networks permit the communications in-between the electronic control units (ECUs) based on the controller area network (CAN). However, IoV system still needs to overcome different challenges before it can perform a substantial contribution. Alongside the challenges, IoV systems requires high security, which made it exposed to different attacks and threats. The attackers launch malicious attacks through the CAN bus risking the security of vehicles. The limited length of CAN packet without authentication or encryption techniques can conduct to many attacks such as DoS, Fuzzy and spoofing attacks. Therefore, to meet the security requirements, the basic process is to employ intrusion detection systems (IDS). These systems aim to identify external vehicle network activity in IoV in order to resolve the aforementioned cybersecurity risks. Recently, researchers brings machine learning (ML) methods as an applied solution for intrusion detection in vehicular system [5]. ML based IDS have a stronger capability for processing large amounts of data and detecting unexpected threats. The main concept is to label the network data into normal data or abnormal data by extracting the data features from the network traffic to train a detection model. The initial ML model is applied when a vehicle is created and leaves the factory [6]. However, when new attacks with new feature distributions have been launched, the model's performance would decrease significantly. To overcome these limitations, transfer learning has been proposed to use data or a model from the source domain to train a machine learning model corresponding to a new task in the target domain. Hence, to fulfill the demands of optimized IoVs intrusion detection systems, in this paper we propose a TRansfer Learning based Intrusion Detection in-vehicle (TRLID) model for IoV using the CAN bus protocol. In addition, We proposed a data preparation method that can effectively clean first the CAN bus data and then transform the data to images to more easily distinguish various cyber-attack patterns. In order to validate and evaluate our method, experiments have been conducted using two in-vehicles datasets, Car-hacking dataset and OTIDS dataset generated from heterogeneous sources that include different types of malicious messages. The experiment results show that the proposed TRLID is efficient for attack detection. The rest of this paper is organized as follows: Section II introduces the related work that uses ML and DL algorithms for vehicle network intrusion detection Section III presents the proposed framework, including data transformation, CNN, transfer learning. Section IV presents and discusses the experimental results. Finally, Section V summarizes the paper.

II. Related Works

In this section, we present some related studies that work on developing an IDS-based ML for IoV. Many researchers worked on different issues from various perspectives in IoV using multiple technologies for security optimization and attacks detection mechanisms. There are also papers proposing new applied solutions on IoVs, or studying the deployment using ML models. Authors in [7] proposed an IDS-based solutions using tree-strucutre ML models for both CAN bus and external attacks detection in IoV. They used the Synthetic Minority Oversampling Technique (SMOTE) for data preprocessing to generate additional data for minority classes with few number of data. The authors used the stacking as ensemble learning approach to improve the accuracy. They trained four machine learning models and integrated their outputs into a meta-learner to create a robust classifier. For DL-based solutions, authors in [9] combined long short term memory (LSTM) and the gated recurrent unit (GRU) models to detect cyberattacks in IoV. The approach is based on different pre-processing methods such as cleaning, shuffling, feature filtering and normalization. These methods are applied to the datasets to improve the performance of the LSTM-GRU model. Meanwhile, authors in [10] designed a specific IDS to identify malicious network activity in In-Vehicle Networks (IVNs), V2V communications and V2I networks using LSTM. The proposed solution consists of three main phases. First, the statistical features are called from both the CAN bus and external network. In the second step, the reduced feature space is given to the recurrent architecture with hidden LSTM layers as sliding temporal windows. After completion of training cycles, minimization of training and validation losses to zero, as well as convergence of weights. It is presumed that the compressed representation of typical traffic has been adequately learned. Finally, the proposed system is evaluated using the car-hacking dataset and the UNSW dataset [11]. In addition, authors in [12] proposed a deep transfer learning based LeCun network (LeNeT) for intrusion detection in the in-vehicle network using the CAN bus protocol. The proposed P-LeNet architecture is made up of seven layers with a total of 12,052 trainable

parameters (weights). The layer is the composition of two convolutional layers, two subsampling layers, one flatten layer, one fully connected layer, and one output layer. The proposed approach is divided into two sections: model training and intrusion detection. First, the selected data was pre-processed, and then used to train the model. Then, the most important parameters have been selected for the model. The suggested model was trained with a dataset that was randomly chosen.

III. Background

A. Attacks On The CAN Bus For The IoV

1) CAN Frame: All nodes connected to the CAN can receive all packet broadcasts. Additionally, a frame in a CAN packet is specified as a structure; it transports a serie of CAN data (bytes) in the network. The arbitration identifier (ID) field in each CAN frame specifies the priority of the transmitted packets. Indeed, when the ID bit value gets lower, the packet's priority become higher. This protocol is designed to prevent collisions on the CAN bus. The CAN data frame consists of four types of standard CAN frame can be identified : The data frame used for data transfer; The remote frame used to allocate a request to the data frame to be transmitted to the target node; Error frame to notify when an error occurs within a delivered frame, and Overload frame to delay the beginning of the next message when the r has not completed processing the message. The CAN frame structure consists of seven fields, as shown in Figure 1.

- Start of frame (SOF): It is consisting of a single dominant bit and alerts all nodes that transmission has started.
- Arbitration Field: It has 11 bits for identifier and one bit for RTR (Remote Transmission Request). During the arbitration procedure, the identifier is treated as a priority, and the RTR is chosen according to the type of CAN frame.
- Control Field: It provides information for the receiver to determine if all packets were successfully received.
- Data field: It is the data used to transmit information from one node to another. It varies from 0 to 8 bytes.
- CRC Field: It ensures the validity of a message as a cyclic redundancy code (CRC).
- Acknowledge Field: It guarantees that the message was received successfully by the receiver node. If the message is valid, the receiver will notify the sender, and change the recessive ACK bit (logic 1) with a dominant bit (logic 0).
- End of Frame: It denotes that the CAN frame has been terminated by a flag with seven recessive bits.

2) CAN Bus Attacks: The in-vehicle network attackers are divided into two types: attackers with physical control of nodes and attackers without physical control of nodes. Attackers that have the physical control to the node can physically alter message transmission, to allow a malicious node to broadcast messages in place of a legitimate node. Nodes without physical access to the node inject malicious messages to influence the operation of the vehicle. In this paper, we consider four attacks as follows (Figure 2):

- DoS attack: an attacker may insert messages with a high priority in a cycle of the bus. DoS attack messages use the highest priority identifier, 0x000, to dominate the bus.
- Fuzzy attack: During Fuzzy attack, an attacker can inject random messages with faked identifiers and data. It can analyze the in-vehicle messages and choose target identifiers to create malicious behaviors.
- Spoofing attack : the attacker attempts to inject a message with a certain CAN ID into the CAN bus in order to cause vehicle abnormalities. Unlike Fuzzy attack, Spoofing attack selects the CAN ID normally broadcast on the CAN bus to attack the network, whereas Fuzzy attack can generate any simulated fake ID to initiate an attack.
- Impersonation Attack: An impersonation attacker can stop the message transmission by controlling the target node and manipulating an impersonating node that can transmit data frames and replies to remote frame like targeted node.

B. Fundamentals of Transfer Learning

As previously indicated, in TL, knowledge acquired in the source domain will be transferred to the target domain to enhance learning for the target task. Therefore, we should provide the definitions of a domain and a task in TL.

First, the two component that make up a domain D are the feature space χ and a marginal probability distribution *P(X)*, where $X = \{x_1, ..., x_n\} \in \chi$ and n represents the number of feature vectors in X. We note *D* = $\{\chi$, *P(X)*$\}$.

A task is defined by two parts, a label space *L* and a decision function *f(.)*. The function *f(.)* is learned from the feature vector and label space pairs $\{x_i, l_i\}$, where the $x_i \in X$ and $l_i \in L$. Generally, the decision function returns the prediction of label $f(x_i)$ given instance x_i. Therefore, the decision function can be written as $f(xi) = P(l_k|x_i)|l_k \in L$, where $k = \{1, ..., |L|\}$. For example, in a binary attack detection, L presents the set of all labels of normal data ("0"), and attack ("1"). Based on instance x in the feature space, $f(.)$ predicts the probability of normal data or attack.

IV. Proposed System

This section presents the overarching workflow of our proposed solutions. We first present the data preparation process, which is the first fundamental step of our system. Then we introduce the established phases of transfer learning.

A. Data Preparation

To reach the best performance and enhance the learning process, the data must be prepared before choosing the TL model. Data preparation involves deleting irrelevant features, converting non-numeric features and removing outliers. To prepare the CAN bus data, we apply two main phases; data cleaning and data transformation.

1) Data Cleaning: The CAN bus dataset is particularly susceptible to inaccuracy and inconsistency. First, Outliers are checking to find out a data point that differs substantially from the rest of the data. In addition, missing data is identified and

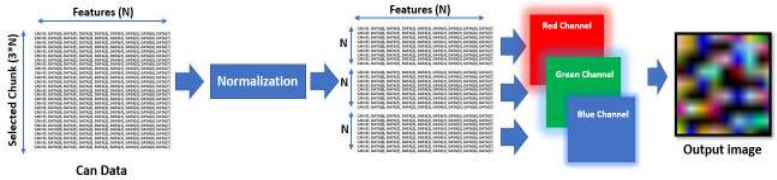


Fig. 3: Data to image transformation

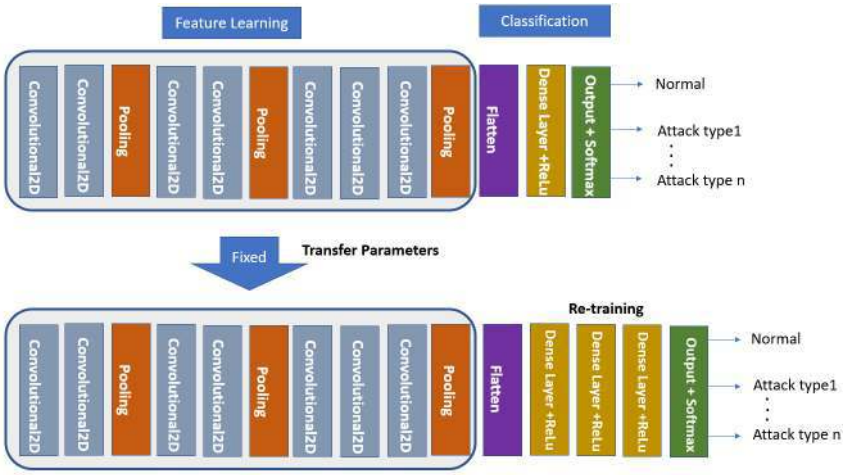


Fig. 4: Representative illustration of the TRLID Architecture

removed. For instance, the qualitative values "Normal" and "Attack" have been altered to "0" and "1" for the dataset's label feature. The *hex2dec* function was used to convert the hexadecimal values of CAN ID feature into decimal values. the Data Field feature of the dataset comprises eight bytes of hexadecimal numbers separated by spaces. The *gsub* function was used to remove spaces between bytes, while the *Rmpfr* function was used to convert hexadecimal values to decimal integers.

2) Data Conversion: The IoV network traffic dataset was recorded in a format other than image, it may be a .csv file, txt, or other type of files. In this section, we discuss the transformation of non-image samples to an image form in order to get effective results for network attack detection. It is essential to transform the CAN signals into images, since the pretrained convolution neural network (CNN) models used for transfer learning are designed to accept 3D image samples as input. we normalize the data collected using:

$$X_{new} = \frac{X - Min(X)}{Max(X) - Min(x)} * 255 \qquad (1)$$

The network data must be normalized into the range of pixel values of images from the range of 0 to 255. After the normalization phase, a chunk of 27 successive samples is selected to convert the data into image with 9*9*3. where 9 is the number of feature in the two datasets. Each 9 chunks are converted in an image matrix channel. Then the three matrix are mapped into RGB channels of an image. The image label depends on the attack patterns on this image. An image is marked as "Normal" if every sample in the image is a normal sample. In fact, if an image includes attack samples, it will be labeled based on the most common attack in this image. Therefore, we discuss the Figure 3 which displays a broad overview of the process.

B. Transfer Learning

Mainly, transfer learning runs first the ID model generation in the source domain. In the next step, the generated ID model update in the target domain after the knowledge transfer. These two domains are described as follows:

- Source domain: The initial step of the transfer learning based architecture consists of the creation of the source domain intrusion model. A source dataset is used to train and validate the IDS model. We use the CNN as the basic model. The CNN's architecture consists of six convolutional layers, three pooling layers to select the most important features and a dense layer. The output layer involves 5 outputs. Each layer takes the previous layer outputs as inputs for the current layer and performs some non-linearities to transform it into a multivariate series whose dimensions are defined by the number of filters in each layer. The structure of the proposed CNN-based model source is illustrated in figure 4. The first layer is the input layer, which is not considered as a network layer because it does not learn anything. The input layer is designed to take the dataset and pass it to the following layer. The convolutional layers are in charge of executing convolution operations. The rectified

linear unit (ReLU) is used with all the convolution layers. The max-poling layers aid in reducing the amount of computing power required to process the data. They are used to select the most important features. The flatten layer converts the pooled feature map to a single column that is passed to the next layer. The fully-connected dense layer reduces drastically the number of trainable parameters in a deep model while enabling the use of a class activation map which allows an interpretation of the learned features. Finally, the output layer carries a number of neurons that corresponds to the number of classes in the dataset. The softmax function is used as the activation function in this layer to predict a probability distribution between normal and attack scenarios.

- Target domain: The second step of the transfer learning process is to apply the source domain knowledge to the target domain. The CNN convolutional base used for the source domain is frozen to avoid the modification of the weights when the model is retrained, and the classifier is trained with its outputs. The CNN used to train the target domain compromises of the frozen layers of the CNN used in the source domain and fully connected layers as an output layer. The ReLU activation function is used in the hidden layers of FC layers. The output layer consists of a FC network with softmax activation.

V. Experimental Results

A. Experiment Setup

The car-hacking dataset comprises CAN packets collected from the OBD-II port. Each CAN packet is defined by three key features: CAN ID which represents the identifier of CAN packet, DATA[0] to DATA[7] which represents the 8 data bytes of the packet, and finally the flag which accepts two values, T and R (T: inject packet and R: normal packet). The dataset includes normal traffic as well as three types of attack: (1) DoS attack: DoS packet with CAN ID =”0X000” is injected every 0.3 milliseconds. (2) Fuzzy attack: Random ID and DATA values are injected every 0.5 milliseconds. (3) Spoofing Attack (RPM/gear): It injects certain CAN ID packets relevant to RPM and gear every 1 millisecond. We use the OTIDS [13] dataset as a target dataset. The OTIDS dataset is also generated by gathering CAN packets via the OBD-II. It includes normal packets as well as DoS attacks with CAN ID of ”0X000”. The CSV files of fuzzy attacks and impersonation attacks do not indicate if a packet is normal or not. The Fuzzy attack injects faked CAN ID and DATA packets with random values. We extract 9 features from CAN packets : CAN ID and the 8 bytes of the packet DATA[0] to DATA[7] from the two datasets.

B. System Metrics

We use several metrics including the accuracy, recall, precision, and FSscore, which are respectively determined by equations. We note also:

- TP: represents the number of correctly assigned positive samples.
- TN: represents the number of correctly assigned negative samples.
- FP: represents the number of incorrectly assigned positive samples.
- FN: represents the number of incorrectly assigned negative samples.

The accuracy is the average proportion of traffic windows accurately allocated to their corresponding class out of the total number of traffic windows.

$$Accuracy = \frac{TP * TN}{TP + FP + TN + FN} \tag{2}$$

The precision is a reliable measurement when the cost of false positives is high. It refers to the usage of advanced technology and methods to pursue high accuracy under the existing physical framework. It is expressed as follows:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

The recall is the ratio of the number of class traffic windows that are successfully allocated to the number of class traffic windows.

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

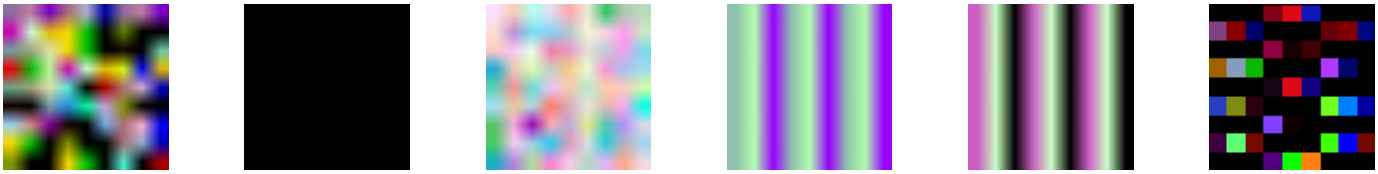
The F1-score is the weighted average of Precision and Recall, taking both FP and FN into account.

C. Hyperparameter Selection for TRLID Model

The selection of the appropriate hyperparameters is very important to improve the model's performance. In our case, many hyper-parameters need to be tuned and optimized to better fit the CNN model. We use the grid search optimization technique to choose the CNN hyper-parameters. We start by defining a search space which includes number of epochs, batch size, learning rate, and dropout rate and optimizer. For each parameter, the grid search was specified. The inputs for the number of epochs and learning rate parameters are 5, 10, 20, 30 and 0.001, 0.01, 0.1, 0.002 respectively. Additionally, we specified 32, 128, 512 and 1024 for the bath size parameters and 0.2, 0.4, 0.5, and 0.6 for the grid search of drop rate. Adam, SGD, Nadam, and Adamax are chosen as inputs of the optimizer search grid. The optimal grid search optimization hyperparameter values are 30 for epochs, 0.001 for the learning rate, 128 for the batch size, 0.4 for the dropout layer, and Adam for the optimizer.

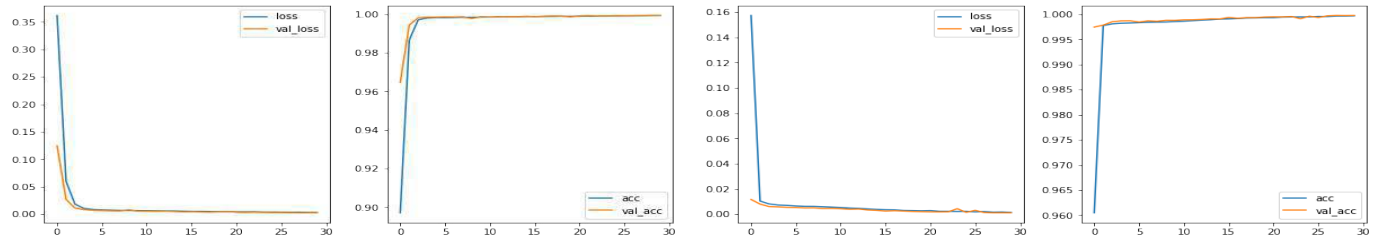
D. Results and Validation

In this section, we will examine the results of the proposed TRLID model. Our transfer learning model is first trained with the car-hacking dataset. Our CNN-based model were trained using OTIDS dataset. As shown, in figure 5, the data transformation method gives attack images with different feature patterns. The DoS attack samples with the highest priority identifier 0x000 are represented by black patterns. The Gear and RPM have a specific patterns resulted by the injection of messages with certain IDs to masquerade



(a) Normal (b) DoS Attack (c) fuzzy Attack (d) Gear Spoofing Attack (e) RPM Spoofing Attack (f) Impersonation Attack

Fig. 5: Image transformation for different types of attacks.



(a) Loss and Accuracy in the training and the validation (Source Domain)

(b) Loss and Accuracy in the training and the validation (Target Domain)

Fig. 6: Loss and Accuracy Plots for The Source Domain and The target Domain

legitimate users. Fuzzy attacks with random messages are more random than normal images. The model accuracy and loss show good results via the successful training of both source domain and target domain. Figure 6a and Figure 6b depict the plot curves of the loss and the accuracy, respectively. Since the curves do not constantly rise past a certain point. We can also observe that the model had not yet overfitted the training for both datasets. The loss plot indicates that the model performs similarly on both the train and test datasets, and that the decreases as the number of epochs increases. In the training of the source domain model, the accuracy achieved 99.91% and the loss reached 0.002%. While in validation, the accuracy reached 99.97% and the loss reached 0.001%. In the other hand, in the training of the target domain model, the accuracy reached 99.83% and the loss value is 0.001%. Meanwhile, in the validation, the accuracy value is equal to 99.87% and the loss value is 0.001%. These results reflect the efficient performance of our proposed model.

VI. Conclusion

The massive quantity of data exchanged between intra-vehicles posed different challenges to traditional IDS. Recently, new proposed IDS are taking benefit of employing DL. They provide an outstanding performance results and good efficiency. Yet, DL still suffer from different limits such as dependent data and lack of labels or annotations. To address the posed limitations, we proposed a deep transfer learning based intrusion detection in-vehicle (TRLID) model for IoV. Our proposed model is built using CNN on car-hacking dataset and been updated with OTIDS dataset. To validate our proposed model, we established an intense experiment. Indeed, by analyzing the obtained efficient results, we concluded that transfer learning provide an ideal solution for CAN bus attacks detection.

References

[1] Nundloll, V., Blair, G., Grace, P. (2009). A component-based approach for (Re)-configurable routing in VANETs. In Proceedings of the 8th International Workshop on Adaptive and Reflective MIddleware (pp. 1–6).

[2] Hasrouny, H., Samhat, A., Bassil, C., Laouiti, A. (2017). VANet security challenges and solutions: A survey. Vehicular Communications, 7, 7–20.

[3] Yang, F., Wang, S., Li, J., Liu, Z., Sun, Q. (2014). An overview of internet of vehicles. China communications, 11(10), 1–15.

[4] Zhou, H., Xu, W., Chen, J., Wang, W. (2020). Evolutionary V2X technologies toward the Internet of vehicles: Challenges and opportunities. Proceedings of the IEEE, 108(2), 308–323.

[5] Al-Jarrah, O., Maple, C., Dianati, M., Oxtoby, D., Mouzakitis, A. (2019). Intrusion detection systems for intra-vehicle networks: A review. IEEE Access, 7, 21266–21289.

[6] Li, X., Hu, Z., Xu, M., Wang, Y., Ma, J. (2021). Transfer learning based intrusion detection scheme for Internet of vehicles. Information Sciences, 547, 119–135.

[7] Yang, L., Moubayed, A., Hamieh, I., Shami, A. (2019). Tree-based intelligent intrusion detection system in internet of vehicles. In 2019 IEEE global communications conference (GLOBECOM) (pp. 1–6).

[8] Li, X., Zhang, H., Miao, Y., Ma, S., Ma, J., Liu, X., Choo, K.K. (2021). CAN Bus Messages Abnormal Detection Using Improved SVDD in Internet of Vehicles. IEEE Internet of Things Journal, 9(5), 3359–3371.

[9] Ullah, S., Khan, M., Ahmad, J., Jamal, S., Huma, Z., Hassan, M., Pitropakis, N., Buchanan, W. (2022). HDL-IDS: a hybrid deep learning architecture for intrusion detection in the Internet of Vehicles. Sensors, 22(4), 1340.

[10] Ashraf, J., Bakhshi, A., Moustafa, N., Khurshid, H., Javed, A., Beheshti, A. (2020). Novel deep learning-enabled LSTM autoencoder architecture for discovering anomalous events from intelligent transportation systems. IEEE Transactions on Intelligent Transportation Systems, 22(7), 4507–4518.

[11] Lokman, S.F., Othman, A., Abu-Bakar, M.H. (2019). Intrusion detection system for automotive Controller Area Network (CAN) bus system: a review. EURASIP Journal on Wireless Communications and Networking, 2019(1), 1–17.

[12] Mehedi, S., Anwar, A., Rahman, Z., Ahmed, K. (2021). Deep transfer learning based intrusion detection system for electric vehicular networks. Sensors, 21(14), 4736.
[13] Lee, H., Jeong, S., Kim, H. (2017). OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame. In 2017 15th Annual Conference on Privacy, Security and Trust (PST) (pp. 57–5709).