Expert Systems With Applications 238 (2024) 122181

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Supervised contrastive ResNet and transfer learning for the in-vehicle intrusion detection system

Thien-Nu Hoang [a,1], Daehee Kim [b,*]

[a] Applied Artificial Intelligence Institute (A^2I^2), Deakin University, Geelong, Australia
[b] Department of Future Convergence Technology, Soonchunhyang University, Asan-si, 31538, Chuncheongnam-do, South Korea

ARTICLE INFO

Keywords:
Controller area network
Intrusion detection
Supervised contrastive learning
Transfer learning

ABSTRACT

High-end vehicles have been equipped with several electronic control units (ECUs), which provide upgrading functions to enhance the driving experience. The controller area network (CAN) is a well-known protocol that connects these ECUs because of its modesty and efficiency. However, the CAN bus is vulnerable to various types of attacks. Although the intrusion detection system (IDS) is proposed to address the security problem of the CAN bus, most previous studies only provide alerts when attacks occur without knowing the specific type of attack. Moreover, an IDS is designed for a specific car model due to diverse car manufacturers. In this study, we proposed a novel deep learning model called supervised contrastive (SupCon) ResNet, which can handle multiple attack classification on the CAN bus. Furthermore, the model can be used to improve the performance of a limited-size dataset using a transfer learning technique. The capability of the proposed model is evaluated on two real car datasets. When tested with the Car Hacking dataset, the experiment results show that the SupCon loss reduces the overall false-negative rates of four types of attack by an average of five times compared to the vanilla cross-entropy loss. In addition, the model achieves the highest F1 score on both the vehicle models of the survival dataset by utilizing transfer learning. Finally, the model can adapt to hardware constraints in terms of memory size and running time to be deployed in real devices.

1. Introduction

The intelligent vehicle industry has gained considerable attention and interest from companies, researchers, and consumers. Many electronic control units (ECUs) are installed inside a smart vehicle to assist with more advanced functions. These ECUs are interconnected through an in-vehicle network (IVN) in which the controller area network (CAN) protocol is widely used, although other technologies, such as the local interconnected network (LIN), FlexRay, and Ethernet, are also available. The CAN protocol lacks encryption and authentication mechanisms despite its fast speed and simplicity. Hence, there is a trade-off between efficiency and security. Many studies demonstrate that a CAN bus network can be attacked in various ways (Hoppe, Kiltz, & Dittmann, 2011; Jo & Choi, 2021; Koscher et al., 2010). Thus, different mechanisms have been introduced to fill the gap in the security of the CAN bus. One of them is developing a system monitoring and detecting attacks in the CAN bus network, which is called an intrusion detection system (IDS). With the rapid development of the machine learning field, various studies applied simple machine learning models, including K-means (D'Angelo, Castiglione, & Palmieri, 2020), K nearest neighbors (KNN) (Derhab, Belaoued, Mohiuddin, Kurniawan, & Khan, 2021), and one-class support vector machine (OSVM) (Avatefipour et al., 2019), as well as complex deep learning models, such as deep neural network (M. J. Kang & Kang, 2016), long short-term memory (LSTM) (Ashraf et al., 2020; L. Kang & Shen, 2021; Taylor, Leblanc, & Japkowicz, 2016), convolutional neural network (CNN) (Ahmed, Ahmad, & Jeon, 2021; M. Chen, Zhao, Jiang, & Xu, 2021; Desta, Ohira, Arai, & Fujikawa, 2022; Hoang & Kim, 2022; Seo, Song, & Kim, 2018; Song, Woo, & Kim, 2020; Sun, Chen, Weng, Liu, & Geng, 2021), and recent transformer networks (Nam, Park, & Kim, 2021; Nguyen, Nam, & Kim, 2023) to build an efficient IDS. Despite good results achieved in previous studies with high detection accuracy and low error rate, there are still two major issues. First, most of them solve the problem of binary classification, which consists of two classes: normal and abnormal. This study aims to solve the multiclass classification problem that distinguishes between normal traffic and different types of attacks. A specific type of attack is then

* Corresponding author.
E-mail addresses: nu.hoang@deakin.edu.au (T.-N. Hoang), daeheekim@sch.ac.kr (D. Kim).
[1] Work done during the Master's degree in Korea.

https://doi.org/10.1016/j.eswa.2023.122181
Received 31 October 2022; Received in revised form 13 October 2023; Accepted 13 October 2023
Available online 16 October 2023

notified to the user by the system. Since multiclass classification is more challenging than binary classification, we propose the concept of contrastive learning (Chopra, Hadsell, & Lecun, 2005). This can improve detection ability by learning from dissimilarity and similarity between training samples. In addition, contrastive learning-based models can solve the challenge of class imbalance in the CAN bus hacking dataset. Second, most machine learning based IDSs learn the behavior of CAN message transmissions of a specific vehicle model. Further, CAN ID meanings are distinct for different vehicle models although the CAN messages follow the same structure. Consequently, if we want to develop an IDS for a newly launched vehicle model, we must collect new large datasets and train a new model. In this study, we apply the transfer learning technique to address the problem.

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

- We introduce a novel model called supervised contrastive (SupCon) ResNet, which combines supervised contrastive learning with the traditional ResNet and minimizes the supervised contrastive loss, rather than the traditional cross-entropy loss. To the best of our knowledge, our study is the first to apply contrastive learning to the in-vehicle IDS.
- We propose using an inductive transfer learning framework, which facilitates the generalization of a deep learning model across different CAN bus data from different car models. We used the pre-trained SupCon ResNet model to transfer learned knowledge from a rich source dataset to other limited target datasets. By this way, the proposed system can save time for data collection on a new vehicle model.
- We prove the efficiency of our proposed system through comprehensive experiments and analyses on two popular real car datasets: the Car Hacking (Song et al., 2020) and Survival (Han, Kwak, & Kim, 2018) datasets. When tested with the Car Hacking dataset, the SupCon loss lowers the overall false-negative rates of four types of attack by almost five times on average, compared the cross-entropy loss. Furthermore, the proposed SupCon-based transfer learning system achieves the highest F1 scores on both KIA Soul and Chevrolet Spark datasets, at 0.9998 and 0.9979 respectively, compared to other baseline models.

After providing an overview of the problem and stating the contributions of the study in Section 1, we introduce relevant background knowledge and security issues in the CAN bus protocol in Section 2. Then, various studies related to the in-vehicle IDS are summarized in Section 3. Our proposed system is described in detail in Section 4. Next, the experimental setup and results are presented in Section 5, where we showed the efficiency of the proposed method empirically. Finally, Section 6 presents the conclusions and directions for future works.

2. CAN bus background

2.1. CAN bus system

The controller area network (CAN) is one of the protocols used in the in-vehicle network comprising many connected ECUs. The CAN bus, which is invented by Robert Bosh GmbH in the early 1980 s, has become common in automotive systems because of its advantages, including high bit rates (up to 1 Mbit/s), cost-effectiveness, and system flexibility (BOSCH CAN Specification Version 2.0, 1991). The concept of CAN protocol comprises message prioritization and multi-master. Every CAN message contains an ID related to the content of the message. Based on this, each ECU obtains its relevant messages from a message filterer. There is no source and destination information in CAN messages. In addition, the ID implies the priority of a message: the lower the ID, the higher the priority. When the bus is idle, any unit can start to transmit a message. The message whose higher priority wins the bus and takes the right to send the message. During transmission, any number of nodes can receive and simultaneously act upon the same message according to the concept of message filtering.

A CAN message is called a frame belonging to four types, such as data, remote, error, and overload frames. While the data frame contains the main information in ECU communication, the remote frame is used when a unit requests peculiar information from other ECUs. Meanwhile, the error frame supports error detection and fault confinement. An overload frame will be sent whenever a node requires a delay of the next frame. Every CAN frame is made up of a sequence of dominant bits (0) and recessive bits (1). The data and remote frames have the same structure, as shown in Fig. 1. The data and remote frames start with a bit called the start of frame (SOF), which is always a dominant bit (0). According to the length of the ID, the CAN data are classified into CAN 2.0A and CAN 2.0B. CAN 2.0A is a shortened version of CAN 2.0B, without the extended identifier. Hence, the CAN 2.0B format structure is described in Fig. 1. The 11-bit length base identifier is followed by a 1-bit substitute remote request (SRR) and 1-bit identifier extension (IDE). The successive 18 and 6 bits represent the extended identifier and control field, respectively. The data field containing 64 bits is the content of the message. The following 16 bits for cyclic redundancy check (CRC) and 2 bits for acknowledgment are utilized in error detection and correction. The end of the frame (EOF) is recognized by seven consecutive bits. For more information regarding the other frame types, we refer to the CAN specification reported by Bosch Robert GmbH (BOSCH CAN Specification Version 2.0, 1991).

2.2. Security in CAN bus protocol

Despite the advantages mentioned above, the most severe problem in CAN protocol is message transmission without authentication and encryption. Since there are no source and destination addresses in a CAN message, any node connected to the bus can obtain the message during message transmission. Consequently, an attacker who compromises a node on the bus can easily sniff the information on the bus. There are two levels of an adversary: weak and strong (Cho & Shin, 2016). A weak attacker can prevent the ECU from transmitting messages or keep the ECU in a listen-only mode. Meanwhile, a strong attacker can completely control an ECU and have access to the memory data. Consequently, malicious messages can be injected to launch the attack in addition to the abilities of the weak attacker. These two levels of attacks are proved in several studies (Hoppe et al., 2011; Jo & Choi, 2021; Koscher et al., 2010). In this study, our proposed IDS handles the strong attacker including three different types of injected messages:

- DoS: The attacker floods the bus by injecting messages containing the highest priority ID 0x0000 and arbitrary data fields. The legitimate messages are prevented from being transmitted, resulting in unusual effects, such as flashing dash indicators, intermittent accelerator/steering control, and even full vehicle shutdown, since the message with the ID 0x0000 always wins the bus (Cho & Shin, 2016).
- Fuzzy: The attacker injects messages with an arbitrary ID and data fields at a high frequency. Aside from being randomly generated, the injected IDs can be chosen from those appearing in the normal traffic, which is supposed to be difficultly detected. The effect of this attack is similar to the DoS attack.
- Target ID: The attacker injects messages with a specific ID and manipulated data fields as his intent. To achieve this, the attack requires the attacker to have knowledge of the meaning of the specific ID by reverse engineering techniques.

3. Related works

3.1. Deep learning-based in-vehicle IDS

Two main approaches to counter in-vehicle cyber-attacks are message authentication and intrusion detection (Cho & Shin, 2016).

SOF	Base Identifier	SRR	IDE	Extended Identifier	Control Field	Data Field	CRC	ACK	EOF
1 bit	11 bits	1 bit	1 bit	18 bits	6 bits	0 to 64 bits	16 bits	2 bits	7 bits

SOF: Start Of Frame SRR: Substitute Remote Request IDE: Identifier Extension
CRC: Cyclic Redundancy Check ACK: Acknowledgement EOF: End Of Frame

Fig. 1. CAN 2.0B data frame format.

Message authentication consumes more resources and time, although it provides a higher security level. Consequently, these approaches decrease the CAN bus performance. Therefore, intrusion detection is preferred. Different types of IDS have been proposed to monitor and analyze transferred messages on the bus, detect vicious behaviors, and make an alert if detected.

With the rapid development of deep learning models, many deep learning-based in-vehicle IDS have been introduced. A study by (M. J. Kang & Kang, 2016) was one of the first research that applied deep learning for CAN bus intrusion detection. The authors developed a deep neural network where each neuron takes each bit of data payload in the CAN message as the input. The network was trained in a supervised manner with two classes: normal and abnormal. However, the proposed model was investigated using a synthetic dataset with simple attack models. Although the result was not impressive, this study is the first to apply advanced deep learning models to the CAN bus network. Several studies applied recurrent neural networks (RNN) to identify injected messages since CAN messages possess sequential patterns. (Taylor et al., 2016) used the long short-term memory (LSTM), which is the improvement of RNN, to predict the next bit in the data payload of normal CAN message sequences. The invasion is detected by checking the difference between the predicted and receiving values on the bus. Meanwhile, the LSTM-based autoencoder adopted by (Ashraf et al., 2020) was trained with special statistic features of the CAN message sequences. These models were designed for a message sequence of a specific CAN ID. This implies that we need to train many models separately, corresponding to the number of IDs in a CAN network of a vehicle. To reduce the number of models, (Nam et al., 2021) employed the bi-directional generative pretrained transformer (GPT) network to predict the next CAN ID in the CAN message sequence, which was used for abnormal detection to identify attacks. Recently, (Nguyen et al., 2023) devised a novel transformer attention-based method that demonstrates impressive performance in the multiple classification task, despite its high complexity.

Aside from time series-based methods, the convolutional neural network (CNN) was exploited in many CAN bus IDS studies. CNN is commonly used for various computer vision tasks, such as object detection, image colorization, and image segmentation. Hence, a CNN-based IDS requires an image representation of CAN messages. In (Seo et al., 2018), multiple one-hot vectors representing consecutive CAN IDs were stacked to form a CAN image, which is fed into the CNN generative adversarial networks (GAN). The proposed method can perform binary classification (normal/abnormal) with an overall accuracy of 97 % and detect unknown attacks. Similarly, a previous study (Song et al., 2020) manipulated CAN ID sequences in binary form, which is used to train a simplified Inception Resnet. The proposed model achieves a low false-negative rate in the case of DoS and spoofing attacks, but the result of the fuzzy attack is not good. Despite the high F1 scores with limited labeled data, the problem is still binary classification. In addition, the combination of CNN and LSTM was utilized in (Sun et al., 2021) to deal with sophisticated attacks. However, the study also solved the binary classification problem. The study (Ahmed et al., 2021) used the knowledge from the VGG-16 model trained on ImageNet datasets to solve the problem of multiple class classification in the CAN bus. Due to the difference in domains between source and target data, the final F1 score is low. In (M. Chen et al., 2021), a GAN trained with an auxiliary task was proposed to deal with multiclass classification and unknown attack detection. A high F1 score of 0.9963 was achieved but training the GAN network is difficult. Meanwhile, (Desta et al., 2022) proposed a concept of recurrence plots, which is the matrix of subtraction values of multiple CAN IDs within a specific window size. In addition, they suggested using a lightweight CNN model with the proposed recurrence plots to foster the speed of IDS. However, the accuracy of multiclass classification is not high. Since multiple-class classification is a complex problem, we are aware of very few studies addressing this topic that have produced significant results.

3.2. Transfer learning for the in-vehicle IDS

All previous models can only be applied to a specific car model. Hence, retraining a new model on a new dataset is required if we build an IDS for a newly released car. It is time-consuming and requires lots of effort to collect data for every newly released car model. Therefore, transfer learning is proposed to address the problem. As far as we know, the study (L. Kang & Shen, 2021) is the first study that employed transfer learning for cross-domain CAN bus data and presented an LSTM-based model to solve the binary classification. The authors tested their proposed scheme on the survival dataset (Han et al., 2018), which is a small dataset. The data from Kia Soul and Chevrolet Spark models were used for testing the transfer learning, while the data of a Hyundai Sonata car was treated as the source dataset which was employed for training a pretrained model. The proposed scheme did not achieve good results, because the amount of source training samples was not enough. Instead, transfer learning is only beneficial when there is a large source dataset. Compared to existing works, this study aims to build a deep learning model that can classify multiple attacks in the in-vehicle network. In addition, we test the transfer learning capability of our proposed model on a large source dataset. As a result, our reported results are more accurate and reliable. In summary, Table 1 compares our proposed scheme to other CAN IDS studies from different perspectives.

3.3. Contrastive learning in IDS

To the best of our knowledge, there are no studies applying contrastive learning in the in-vehicle IDS design. However, some recent works have proposed the use of contrastive learning to the network IDS (Andresini, Appice, & Malerba, 2021; Liu, Wang, Jia, Luo, & Wang, 2022; Lopez-Martin, Sanchez-Esguevillas, Arribas, & Carro, 2022). For example, (Andresini et al., 2021) combined the autoencoder and triplet loss to demonstrate an IDS on various network datasets, such as KDDCUP99,[2] AAGM17,[3] and CICIDS17.[4] Meanwhile, autoencoder-based contrastive learning was introduced as a part of a multi-task model in (Liu et al., 2022). In addition, (Lopez-Martin et al., 2022) suggested a novel concept of contrastive learning, in which they projected the labels and features into the same representation space. In the

[2] https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
[3] https://www.unb.ca/cic/datasets/android-adware.html.
[4] https://www.unb.ca/cic/datasets/ids-2017.html.

Table 1
Deep Learning CAN-IDS Summary.

Related work	Year	Main methods	Features	Multiclass classification	Real time evaluation	Real dataset	Transfer learning
(M. J. Kang & Kang, 2016)	2016	DNN	Data payload				
(Taylor et al., 2016)	2016	LSTM	Data payload			√	
(Seo et al., 2018)	2018	CNN GAN	CAN ID			√	
(Ashraf et al., 2020)	2020	LSTM Autoencoder	CAN ID			√	
(Song et al., 2020)	2020	CNN	CAN ID		√	√	
(L. Kang & Shen, 2021)	2021	LSTM	CAN ID			√	√
(Nam et al., 2021)	2021	Bi-directional GPT	CAN ID			√	
(Sun et al., 2021)	2021	CNN-LSTM	CAN ID		√	√	
(Ahmed et al., 2021)	2021	CNN	Data payload	√		√	√
(M. Chen et al., 2021)	2021	CNN GAN	CAN ID	√	√	√	
(Hoang & Kim, 2022)	2022	Convolutional Adversarial Autoencoder	CAN ID		√	√	
(Desta et al., 2022)	2022	CNN	CAN ID	√	√	√	
(Nguyen et al., 2023)	2023	Transformer Attention	CAN ID + Data Payload	√	√	√	√
Ours		CNN with Supervised Contrastive Learning	CAN ID	√	√	√	√

classification phase, the predicted label is a class having the closest distance to the input features in that representation space. All studies mentioned suggested that contrastive learning is suitable to solve the problem of class imbalance, which occurs in in-vehicle IDS research.

3.4. Semi-supervised learning based IDS

One line of research closely aligned with our study involves semi-supervised learning (SSL) based IDS. SSL methods address the issue of limited labeled data by leveraging abundant unlabeled data to map information into a latent space, thereby enabling effective classification within that space (K. Chen, Rögnvaldsson, Nowaczyk, Pashami, Johansson, & Sternelöv, 2022; N. Doulamis & Doulamis, 2014; Mammeri, Zhao, Boukerche, Siddiqui, & Pekilis, 2019). SSL approaches can be categorized into four main streams: generative models, self-training-based methods, co-training and multiview learning, and graph-based methods (Zhu, 2005). For more detail review, we refer to the SSL survey in (Zhu, 2005). In the context of CAN-bus intrusion detection, the pioneering work of (Hoang & Kim, 2022) applied SSL to confront the challenge of scarce data. Specifically, their proposed model integrates autoencoders and generative adversarial networks in a two-stage training process: the initial stage focuses on learning robust data representations, which subsequently enhance the performance of supervised learning in the subsequent stage. Additionally, they achieve commendable results in classifying unknown attacks.

Both contrastive learning and SSL-based methods share the objective of discovering highly generalized data representations that contribute to effective classification tasks. However, these approaches stem from distinct perspectives. While SSL-based methods capitalize on unlabeled data, contrastive learning methods glean insights from comparisons across diverse classes. It is worth emphasizing that the primary advantage of our proposed model in this paper, compared to SSL-based methods, lies in its ability to leverage label information from available data sources without necessitating additional unlabeled data. Through the integration of label information into the contrastive learning framework, our model enhances the classification task and cultivates data representations that exhibit greater generalization for the purpose of transfer learning.

4. Methodoloy

4.1. Problem formulation

The intrusion detection system is installed in the CAN bus to monitor the CAN network and provide an alert if there is any malicious message. By using the fluctuation in the CAN ID sequence of consecutive CAN messages, this study aims to build an intrusion detection system f that classifies a sequence of CAN IDs into a set of C attack types, denoted as $\mathscr{A} = \{a_0, a_1, a_2, .., a_C\}$, where a_0 is a normal sequence. Suppose that several CAN log messages are collected from a car model under both normal and attack circumstances. After processing, the data $D = \{(\boldsymbol{X}_i, y_i)\}$ are ready to train a machine learning model for the intrusion detection task, where $\boldsymbol{X}_i$ is a sequence of CAN IDs and $y_i \in \mathscr{A}$ is the corresponding label. The objective is to build a multiple class classification IDS $f(\cdot|\boldsymbol{X})$, which determines whether a CAN IDs sequence $\boldsymbol{X}_i$ belongs to a normal class or one of C predefined attack types as accurately as possible.

The most important challenge for a supervised-based IDS is how to collect enough attack samples, especially for a newly released car model. We cannot apply a model trained on a CAN dataset A to another CAN dataset B if A and B are from different manufacturers because each car model owns a specific CAN IDs pattern. Transfer learning can address this problem by extracting meaningful knowledge from a pre-trained model. We assume that an elegant source dataset D_s exists collected from a car model s. From this, we can build an efficient IDS $f_s(\cdot|\boldsymbol{X}_s)$ for the source car. The final objective is to build an IDS model for the target model $f_t(\cdot|\boldsymbol{X}_t, f_s)$, with a limited target dataset D_t where $|D_t| \ll |D_s|$.

4.2. Proposed system

The proposed system presented in Fig. 2. Overview of the proposed system. comprises two stages: supervised contrastive learning from the source data and transfer learning for the target data. While supervised contrastive learning reduces the effect of class imbalance in the training data, transfer learning addresses the problem of limited data on a newly released car model.

In the first stage, we collect and gather the data from data stream, which is then used to build the CAN ID frame. After being labeled, the preprocessed CAN ID frames from source data are fed into the SupCon

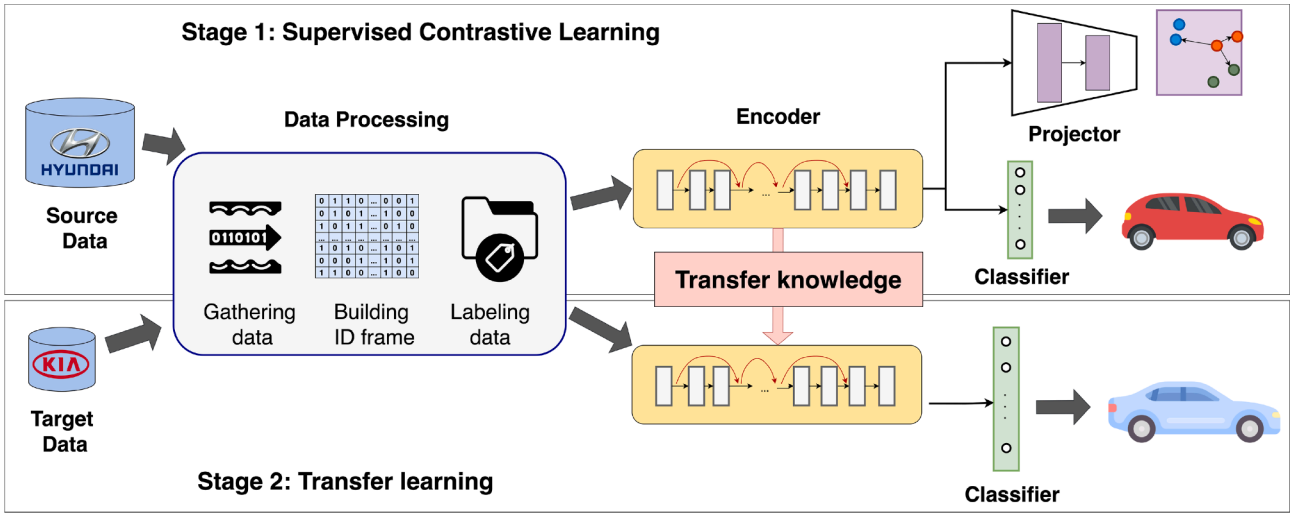

Fig. 2. Overview of the proposed system.

ResNet model. The proposed SupCon ResNet model includes three parts: the encoder, projection, and classification networks.

- The encoder network (Encoder) maps the input $\boldsymbol{X}$ into the representation feature space. In this study, we designed a compact ResNet-18 architecture used as the Encoder.
- The projection network (Projector) projects the outputs from the encoder network into another space, which is normalized and used to measure the distance for the supervised contrastive loss. From experiments, the model with a projector produces a higher accuracy, compared to the model without it (T. Chen, Kornblith, Norouzi, & Hinton, 2020). In addition, a non-linear projection network is assumed to be a better choice than the linear one. Hence, a multiple layer perception (MLP) with one hidden layer is used in our model.
- The classification network (Classifier) is a linear layer performing the final classification task that maps the outputs from the Encoder to $\boldsymbol{C}$ classes.

The Classifier is trained with traditional CE-entropy loss thereafter, while the Encoder and Projector are trained together with SupCon loss.

In the second stage, we performed data processing on the target data the same as the source data. Training the target model from scratch may lead to the overfitting problem because of the small size dataset. Hence, we utilized the pre-trained model f_s for fine-tuning the target model. After completing the training of the source model, we transferred the weights of the encoder from the source model to the target model. After training the classifier on top of the frozen encoder, the entire model is trained with a small learning rate and the unfrozen encoder. The outcome of the system is the IDS model composed of the encoder and classification networks, which will be deployed in a real environment.

4.3. Data preprocessing

In this study, only CAN IDs in the CAN messages were used, which was motivated by the message transmission mechanism in the CAN bus protocol (Song et al., 2020). Each CAN ID in the CAN bus has its message transmission cycle. Meanwhile, the message is broadcasted on the CAN bus based on the priority mechanism. Therefore, we believe that a certain sequential pattern exists in the CAN ID sequence. If an attacker injects some messages into the CAN bus, the pattern will be broken. We stacked N sequential CAN IDs representing 29-bits, to create a CAN ID sequence frame. We choose $N = 29$ because of the convenience of processing square matrix in the CNN architecture network. Consequently, the input fed into the proposed model is a matrix of 29×29, which is visualized in Fig. 5.

4.4. ResNet architecture

The ResNet (Residual Network), which is introduced first in 2015 by (He, Zhang, Ren, & Sun, 2015), has become a popular architecture in deep learning. This allows us to train a much deeper network without performance degradation. Theoretically, the more layer, the better result a deep learning model provides. However, training a deep network is challenging because of vanishing gradient and saturated training error. In a previous study regarding ResNet, the authors introduced the concept of skip connection to solve the problems. At that time, the 152-layer ResNet architecture outperformed the other well-known models, such as AlexNet and VGG, to win first place in the 2015-ILSVRC competition.

The Fig. 3 displays a residual block, the core element in the ResNet. The block takes the input $\boldsymbol{x}$ and outputs $\mathbf{y}$ as follows:

$$\boldsymbol{y} = \mathcal{F}(\boldsymbol{x}, \{\boldsymbol{W}_i\}) + \boldsymbol{x} \tag{1}$$

where the function $\mathcal{F}(\boldsymbol{x}, \{\boldsymbol{W}_i\})$ represents the residual mapping to be learned with the weight of layers $\boldsymbol{W}_i$. The $\boldsymbol{x}$ identity mapping is called a "shortcut connection" that is element-wise addition to the function $\mathcal{F}$. Consequently, the simple concept of adding the identity $\boldsymbol{x}$ to the outcome does not increase the total number of parameters, depth, width, and computational cost (except for the negligible element-wise addition), compared to the plain network. The authors also showed that ResNet, which was 20 and 8 times deeper than AlexNet and VGG respectively, still has lower complexity.

The CAN bus IDS requires not only high accuracy but also low running time. Therefore, we design a compact ResNet-18 to meet these requirements. The Fig. 4 shows the detail of our architecture used for the encoder network. Concretely, we reduced the number of channels of each layer in the original ResNet-18 architecture by four times to produce the compact version. After passing through the convolutional layer with a kernel size of three, the sample goes through eight residual blocks to produce a hidden features representation with the size of 128.

Algorithm 1. Supervised Contrastive Training.

Input:

- $\mathscr{D}$: a dataset includes $\{X_i, y_i\}$ where $0 \leq i < N$, N is the total number of samples, $X_i \in R^{29\times29}$, $y_i \in [0, C)$.
- *n_epochs, learning_rate, class_epoch*: the hyperparameters for training including the number of epochs for training, the initial learning rate, and the epoch to start training the classifier network, repspectively.

Output:

- $\theta_{Enc}, \theta_{Proj}, \theta_{Classifier}$: the weights for encoder, projector, and classifier networks, respectively.

for *epoch* from 1 to *n_epochs* do

Adjust the *learning_rate* with cosine annealing.

Update θ_{Enc} and θ_{Proj} by optimizing the SupCon loss

$$\mathscr{L}_{supcon} = -\sum_{i=1}^{N}\frac{1}{N_{y_i}-1}\times\sum_{j=1}^{N}1_{i\neq j}1_{y_i=y_j}\log\left(\frac{\exp\left(z_i\cdot\frac{z_j}{\tau}\right)}{\sum_{k=1}^{N}1_{k\neq i}\exp\left(z_i\cdot\frac{z_k}{\tau}\right)}\right),$$

where $z_i = f_{Proj}\left(f_{Enc}(X_i, \theta_{Enc}), \theta_{Proj}\right)$.

if *epoch* >= *class_epoch* then

Update $\theta_{Classifier}$ by optimizing the cross-entropy loss

$$\mathscr{L}_{CE} = -\sum_{i=1}^{N}y_i\log\left(\frac{e^{f^c_{Classifier}\left(X_i,\theta_{Classifier}\right)}}{\sum_j^C e^{f^j_{Classifier}\left(X_i,\theta_{Classifier}\right)}}\right).$$

end if

end for

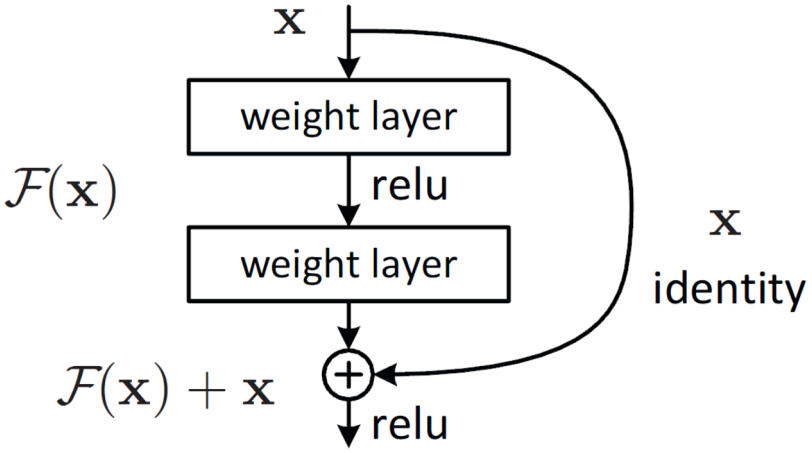

Fig. 3. 3 The residual block in ResNet.

4.5. Supervised contrastive model

Compared to the traditional cross entropy that learns from the label, contrastive learning learns from the dissimilarity between samples. In general, the goal of contrastive learning is to produce embedding features wherein similar samples stay close to each other, while dissimilar ones are distant from each other. The original contrastive loss attempts to minimize the embedding distance when they are from the same class by taking a pair of (x_i, x_j) as the input; otherwise, it maximizes the distance:

$$\mathscr{L}_{cont}(x_i, x_j, \theta) = 1[y_i = y_j]\left\|f_\theta(x_i) - f_\theta(x_j)\right\|_2^2 + 1[y_i \neq y_j]max(0, \epsilon - \left\|f_\theta(x_i) - f_\theta(x_j)\right\|_2^2). \quad (2)$$

where y_i, y_j are the labels of x_i and x_j respectively; $f_\theta(\cdot)$ is the encoder function; and ϵ is a hyperparameter defining the lower bound distance between samples of different classes. Contrastive loss can be better than the cross-entropy if it is trained with hard positive/negative pairs. For example, a hard positive pair comprises two samples that belong to the same class but appear different. Meanwhile, a hard negative pair includes those from different classes but appear quite similar. The topic of how to design an efficient hard mining technique for effective training is also a promising direction in contrastive learning research. In 2020, the SupCon loss (Khosla, P., Teterwak, P., Wang, C., Sarna, A., Research, G., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D. (2020), 2020) was proposed to solve the problem of the original contrastive loss. It outperformed the traditional cross-entropy on the ImageNet datasets in terms of the accuracy. The SupCon loss is extended from the contrastive learning in a self-supervised manner (T. Chen et al., 2020) by leveraging the label information. Fig. 5 illustrates the structure of the supervised contrastive model, compared to conventional neural networks trained

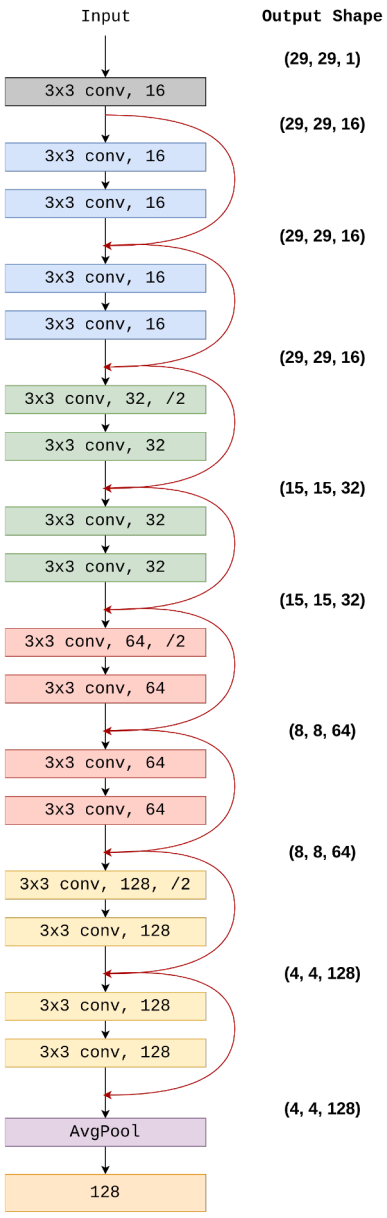

Fig. 4. Detail of the compact ResNet encoder network.

with cross-entropy loss. The SupCon architecture has three main parts: the encoder network, projector network, and classifier. The encoder network maps the input $\boldsymbol{x}$ to a representation vector, $\boldsymbol{r} = Enc(\boldsymbol{x})$. After being normalized, $\boldsymbol{r}$ is fed into the projector network to produces the representation variable $z = Proj(\boldsymbol{r})$. The encoder and projector network are trained with the supervised contrastive loss as below:

$$\mathscr{L}_{supcon} = -\sum_{i=1}^{N} \frac{1}{N_{y_i} - 1} \times \sum_{j=1}^{N} 1_{i \neq j} 1_{y_i = y_j} log\left(\frac{exp\left(z_i \cdot \frac{z_j}{\tau}\right)}{\sum_{k=1}^{N} 1_{k \neq i} exp\left(z_i \cdot \frac{z_k}{\tau}\right)}\right) \tag{3}$$

where N is the total number of training samples, N_{y_i} is the number of positive samples that have the same label y_i with the sample i, j and k are the indices of positive samples and all samples in the training set, respectively. In addition, temperature parameter τ controls the smoothness of probability distribution. A small τ will be good for training, but too small τ can lead to unstable training because of numerical instability. In the original paper, the authors state that $\mathscr{L}_{supcon}$ possesses the implicit property that encourages the hard positive/negative mining without performing it explicitly. As a result, the learning process for the supervised contrastive model does not need to perform hard mining, which slows down the training time. For more detail, we refer to the proof of the original paper (Khosla, P., Teterwak, P., Wang, C., Sarna, A., Research, G., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D. (2020), 2020). By learning from dissimilarity/similarity, the model produces good representations $\boldsymbol{r}$, which are then used to train the classifier with cross-entropy loss as usual. The training process is summarized in Algorithm1 where the SupCon and cross entropy losses are jointly optimized. Two main steps are involved: training the encoder and projector, then training the classifier. To be more specific, we trained the classifier with the encoder, which was trained through several epochs previously by the SupCon loss. This improves classification accuracy since it is fed with robust latent representations as a result of contrastive learning. For learning rate schedule, we also applied cosine annealing (Loshchilov & Hutter, 2016), which is suggested to boost the convergence of the optimization algorithm.

4.6. Transfer learning

Transfer learning is a technique for improving the performance of a target model on a limited target data by utilizing knowledge from a related source model, trained on an abundant source data. According to (Pan & Yang, 2010), the formal definition of transfer learning is described as follows:

- Domain: A domain $\mathscr{D} = \{\mathscr{X}, P(\boldsymbol{X})\}$ includes two components: a feature space $\mathscr{X}$ and a marginal distribution $P(\boldsymbol{X})$ of data $\boldsymbol{X}$.
- Task: A task $\mathscr{T} = \{\mathscr{Y}, f\}$ comprises a label space $\mathscr{Y}$ and a predictive function $f = P(\boldsymbol{Y}|\boldsymbol{X})$, which is trained on a dataset consists of pairs $\{\boldsymbol{x_i}, y_i\}$, where $\boldsymbol{x_i} \in \boldsymbol{X}$ and $y_i \in \boldsymbol{Y}$.
- Transfer learning: Given a source domain $\mathscr{D}_S$, a corresponding source task $\mathscr{T}_S$, and a target domain $\mathscr{D}_T$ and target task $\mathscr{T}_T$, the objective of transfer learning is to learn the target conditional probability distribution $P(\boldsymbol{Y_T}|\boldsymbol{X_T})$ in $\mathscr{D}_T$ with the information gained from $\mathscr{D}_S$ and $\mathscr{T}_S$. Here, $\mathscr{D}_S \neq \mathscr{D}_T$ or $\mathscr{T}_S \neq \mathscr{T}_T$. The assumption is that there are enough labeled source examples, which are substantially larger than labeled target examples.

Based on the relationship between the source and target domains, source and target tasks, and availability of data, transfer learning techniques have various types: inductive, transductive, and unsupervised. For more details on these categories, we refer to the survey of transfer learning (Pan & Yang, 2010).

We have the $\mathscr{D}_S$ and $\mathscr{D}_T$ domains from different car models to apply the transfer learning definition to the IDS for the CAN bus. Although they have the same data format that follows the predefined CAN message structure, each dataset owns a different message transmission behavior. This implies the $\mathscr{X}_S = \mathscr{X}_T$, whereas $P(\boldsymbol{X_S}) \neq P(\boldsymbol{X_T})$. These properties prevent the ability of a generalized IDS model for all car models. In practice, it is difficult to collect a large amount of training data for each car model. Based on the framework of (Pan & Yang, 2010), the inductive transfer learning technique is suitable for our problem. We utilized the encoder in the supervised contrastive learning model, which is trained on a source data, as a pre-trained model. The predictive target model includes the encoder that has the same structure as the source model and top classifier, which is attached to solve the target task. First, we trained the target model with the frozen encoder. This step can be considered as the initialization of the weights of the classifiers. Finally, we trained the entire model after unfreezing the encoder network with a small learning rate to avoid overfitting, which is called fine-tuning stage.

5. Experimental results

5.1. Experiment setup

We used two popular datasets produced by the Hacking and Coun-

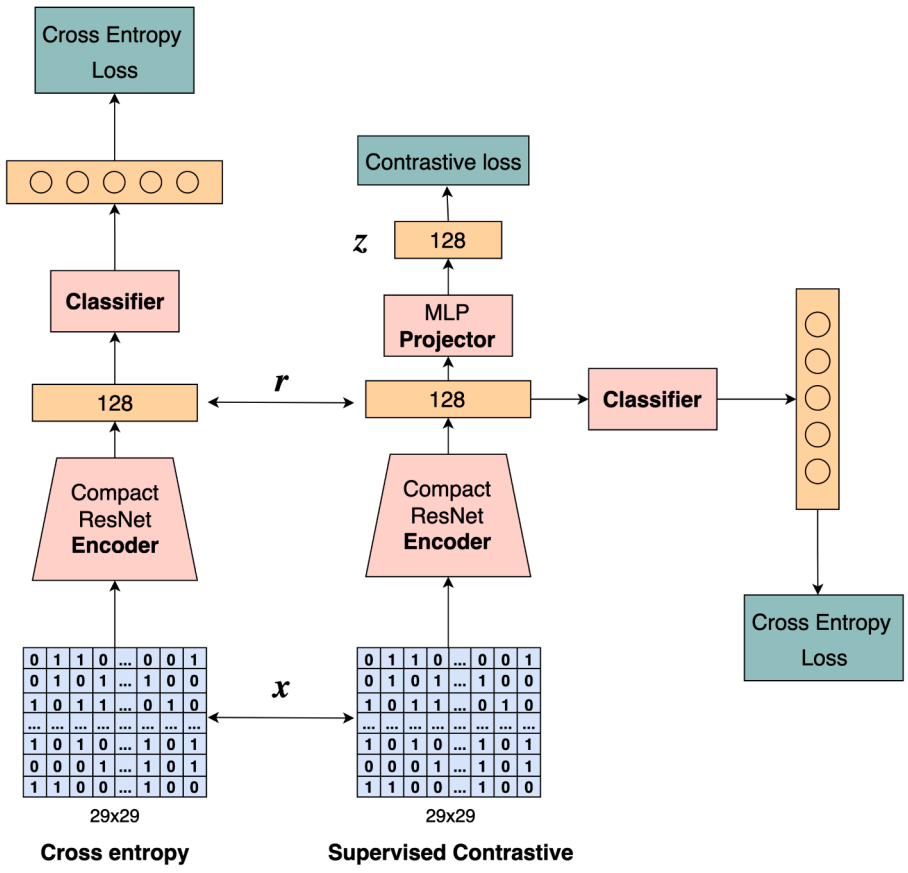

Fig. 5. Difference between the SupCon ResNet and traditional CE ResNet.

termeasure Research Lab (HCRL) of Korea University: Car Hacking (Song et al., 2020) and Survival (Han et al., 2018) datasets. The Car Hacking dataset was collected from the Hyundai Sonata model, while the Survival dataset was collected from three different car models (i.e., Hyundai Sonata, KIA Soul, and Chevrolet Spark). We used the data from the Sonata Hyundai car model in the Car Hacking dataset as the source and treated the others as the target ones since the data of the Hyundai Sonata car model is much larger than the others. The label space of source and target tasks are different: $\mathcal{Y}_S \in \mathbb{R}^5$ including normal, DoS, fuzzy, spoofing RPM, and spoofing gear information. Meanwhile, $\mathcal{Y}_T \in \mathbb{R}^4$ including normal, DoS, fuzzy, and malfunction. In detail, Table 2 lists the data size information of each dataset and Fig. 6 illustrates the distribution of classes in Car Hacking and Survival datasets.

Each message in both two datasets follows the same format containing a timestamp, CAN ID in HEX, the number of data bytes, 8 bytes of data, and the label. We extracted CAN IDs and transformed them from hexadecimal to 29-bit representation. Then, the CAN ID sequence frame was built by stacking 29 sequential samples together. A stride value s is the number of messages between two continuous frames. The stride value affects the number of frames. For example, there are more frames created if the stride value is small, but there are fewer variations between them. We choose $s = 15$ for the Car Hacking dataset and $s = 10$ for the Survival dataset because the size of the Survival dataset is quite smaller than that of the Car Hacking dataset.

The dataset comprises multiple files corresponding to the attack types. For each file, the frame was labeled as normal if there is no injected message, whereas the label of the frame is a non-zero integer number considering the type of attack. We combined all the processed frames to obtain the final dataset. The train/test splitting rate is 7:3. The final number of training and testing samples are summarized in Table 3. We trained the model on the training set and evaluated it on the test set. For each model, we performed the entire process from dataset splitting to evaluation five times and reported the average results.

The experiments were conducted on a server provided with 32 Intel (R) Xeon(R) Silver 4108 CPUs @ 1.80 GHz, a memory of 128 GB, and an Nvidia Titan RTX 24 GB GPU. All the code written in Python 3.7.10 and Pytorch 1.9.0 is published on the Github.[5]

5.2. Evaluation metrics

As our problem is multiclass classification, we evaluated the proposed model using the false-negative rate (FNR), recall (Rec), precision (Prec), and F1 score (F1). These metrics were calculated from the confusion matrix using true/false positive/negative samples. When we calculated a metric of a specific class, this class is considered positive, whereas the others are considered negative. Concretely, the FNR is the fraction between the false-negative and the total number of positive.

$$FNR(\%) = \frac{FN}{TP + FN} \times 100. \tag{4}$$

In the case of the IDS, the FNR must be as small as possible. In addition, the FNRs of the attack classes are expected to be smaller than that of the normal class because the consequence of a missed detected attack sample is more dangerous than that of a false alarm alert. Meanwhile, the recall is calculated as the fraction between true positive and the total sum of true positive and false negative.

$$Rec = \frac{TP}{TP + FN}. \tag{5}$$

The precision is calculated as the fraction between the true positive

[5] Source code is available at https://github.com/htn274/CAN-SupCon-IDS.

Table 2
The number of messages in Car Hacking and Survival datasets.

Attack type	#Messages (Hyundai Sonata)	#Messages (KIA Soul)	#Messages (Chevrolet Spark)
DoS Attack	3,078,250 587,521	181,901 33,141	120,570 22,587
Fuzzy Attack	3,347,013 491,847	24,990 39,812	65,665 5,812
Malfunctioning	– –	173,436 7,401	79,787 8,047
Gear Spoofing	4,443,142 597,252	– –	– –
RPM Spoofing	4,621,702 654,897	– –	– –

Note: The Sonata values come from the Car Hacking dataset, while the other values are from the Survival dataset. The red color indicates injected messages, while the black indicates normal messages.

and the total sum of true and false positives.

$$Prec = \frac{TP}{TP + FP}. \tag{6}$$

For our problem, the proposed model should achieve both high recall and high precision. Therefore, the F1 score is used to balance these two metrics. The F1 score is defined as the harmonic mean of precision and recall.

$$F1 = 2 \times \frac{Prec \times Rec}{Prec + Rec}. \tag{7}$$

5.3. Choosing hyperparameters for the SupCon ResNet model

Choosing the right hyperparameters is important in deep learning because it can boost the performance of a model. Three hyperparameters should be considered in the SupCon ResNet model: the learning rate, batch size, and τ value. For simplicity, we set the τ value as 0.07 as advised in the original paper (Khosla, P., Teterwak, P., Wang, C., Sarna, A., Research, G., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D. (2020), 2020) and tuned only the learning rate and batch size. The authors of the SupCon study stated that the SupCon loss takes advantage of large batch sizes. Hence, we tried different batch sizes of 512, 1024, 2048, and 4096. We set the learning rate as 0.05 for 512 and 1024 batch sizes. Meanwhile, the learning rate of 0.1 was used for the model with the batch size of 2048 and 4096. Finally, we trained the top classifier for four models with the same configurations with batch sizes of 256 and a learning rate of 0.01 after 150 epochs. In addition, we also applied the cosine annealing for learning rate schedule. We reported the FNR grouped by attack types of four models in Fig. 7. The results showed that the classifier trained with SupCon loss classifies normal and attack samples well on the Car Hacking dataset, as all models have FNRs lower than 0.1 %. The Fuzzy attack has the highest FNR in almost settings, while the DoS attack is easily detected. This finding is aligned with the study in (Song et al., 2020). Moreover, we compared these models in terms of the average FNRs. The results suggest that the larger batch size does not show better results. Overall, the model with a batch size of 512, which is the best model with the lowest mean of FNRs, is chosen for the following experiments.

Table 3
The number of train and test samples for the Car Hacking and Survival datasets.

Dataset	Training (#Samples)	Testing (#Samples)
Car Hacking	773,235	331,391
Survival – KIA Soul	42,366	18,160
Survival - Chevrolet Spark	18,614	7,981

5.4. Comparison with other supervised methods

To show the advantage of the supervised contrastive loss, we compared the SupCon ResNet with the vanilla CE ResNet, which also utilized the compact ResNet architecture. The main difference is the way these models are trained, as described in Section 4.5. In addition, we compared the proposed model to various recent supervised-based CAN-bus IDSs including:

- Histogram-based KNN (Derhab et al., 2021) constructs histograms summarizing the latent patterns in CAN ID sequences within a specific window size. The histograms are then inputted to a simple classifier – KNN with the recommended optimal window size of 30 as suggested in the study.
- Inception ResNet (Song et al., 2020) and Rec-CNN (Desta et al., 2022) are two state-of-the-art CNN based methods in CAN IDSs. Since the original studies addressed the binary classification problem, we re-implemented the models in a multiple class classification to compare with our model.
- LSTM and Transformer Attention based approaches (Nguyen et al., 2023) are time-series-based CAN IDSs. We report the best results of these models with the optimal window size of 128.

As we can see from Table 4, our proposed ResNet architectures (both CE ResNet and SupCon ResNet) yields smaller FNRs of all attacks compared to most studies. The results of the Inception ResNet and histogram based KNN are worse than those reported in the original papers. This indicates that multiple class classification is therefore more challenging than binary classification. Unexpectedly, the performance of LSTM model is lower than other CNN-based methods. In addition, the Transformer Attention based model demonstrates comparable results, particularly for DoS and Fuzzy attacks, with the FNRs of 0.0 % and 0.1 %, respectively. However, it is worth nothing that the optimal window size of the Transformer Attention based model is 128 CAN frames, which is significantly larger than that our window size of only 29 frames. Furthermore, the Transformer Attention based model introduces high complexity which will be discussed in Section 5.7.

Compared to the CE ResNet, the SupCon loss reduces the FNRs almost five times on average. As a result, the proposed SupCon model achieves the highest average F1 score of 0.9998 for all the attacks. In addition, Fig. 8 illustrates the confusion matrices of the two models: the

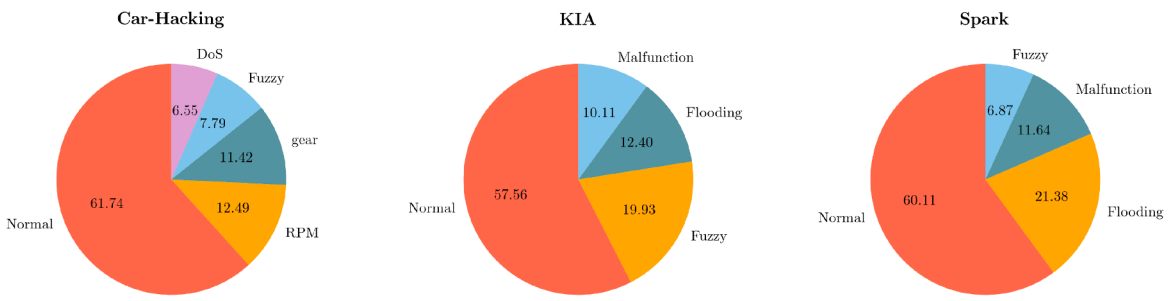

Fig. 6. The distribution of classes in Car Hacking and Survival datasets.

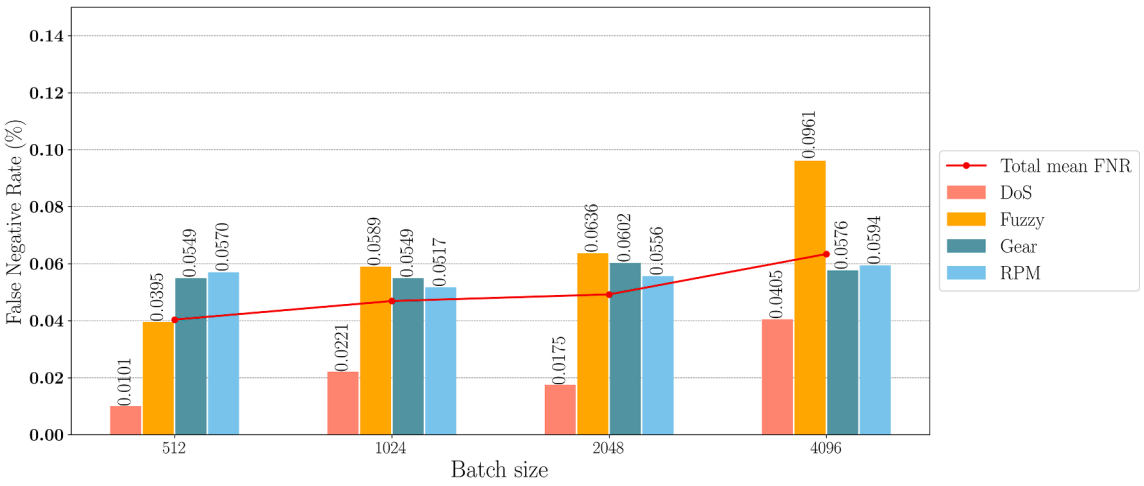

Fig. 7. False negative errors grouped by attack types of each set of hyperparameters.

Table 4
Comparison of the proposed SupCon model and other supervised models.

Attack Type	Model	FNR	Rec	Prec	F1
DoS	Inception ResNet (Song et al., 2020)	0.30	0.9970	0.9999	0.9985
	Histogram KNN (Derhab et al., 2021)	0.51	0.9950	0.9997	0.9973
	Rec-CNN (Desta et al., 2022)	2.0	0.9800	0.9924	0.9862
	LSTM (Nguyen et al., 2023)	0.74	0.9926	0.9911	0.9919
	Transformer (Nguyen et al., 2023)	**0.0**	**1.0**	**1.0**	**1.0**
	CE ResNet (Ours)	0.21	0.9979	0.9997	0.9987
	SupCon ResNet (Ours)	0.02	0.9998	1.0	0.9999
Fuzzy	Inception ResNet (Song et al., 2020)	0.33	0.9967	0.9992	0.9979
	Histogram KNN (Derhab et al., 2021)	1.20	0.9880	0.9998	0.9939
	Rec-CNN (Desta et al., 2022)	17.0	0.8300	0.8964	0.8619
	LSTM (Nguyen et al., 2023)	3.27	0.9673	0.9666	0.9670
	Transformer (Nguyen et al., 2023)	**0.01**	**0.9999**	0.9999	**0.9999**
	CE ResNet (Ours)	0.24	0.9976	0.9997	0.9987
	SupCon ResNet (Ours)	0.06	0.9994	**1.0**	0.9997
Gear Spoofing	Inception ResNet (Song et al., 2020)	0.20	0.9980	0.9996	0.9988
	Histogram KNN (Derhab et al., 2021)	0.23	0.9977	**1.0**	0.9989
	Rec-CNN (Desta et al., 2022)	7.63	0.9237	0.8244	0.8763
	LSTM (Nguyen et al., 2023)	3.12	0.9688	0.9983	0.9833
	Transformer (Nguyen et al., 2023)	0.32	0.9968	0.9938	0.9953
	CE ResNet (Ours)	0.11	0.9989	0.9997	0.9993
	SupCon ResNet (Ours)	**0.06**	**0.9994**	0.9999	**0.9997**
RPM Spoofing	Inception ResNet (Song et al., 2020)	0.19	0.9981	0.9997	0.9989
	Histogram KNN (Derhab et al., 2021)	0.16	0.9984	**1.0**	0.9992
	Rec-CNN (Desta et al., 2022)	15.50	0.8450	0.8763	0.8604
	LSTM (Nguyen et al., 2023)	2.50	0.9749	0.9553	0.9650
	Transformer (Nguyen et al., 2023)	**0.06**	**0.9994**	0.9999	0.9996
	CE ResNet (Ours)	0.12	0.9988	0.9998	0.9993
	SupCon ResNet (Ours)	**0.06**	**0.9994**	**1.0**	**0.9997**

SupCon ResNet and CE ResNet to show the superiority in detection performance of using the SupCon loss. Noticeably, the SupCon ResNet decreases misdetected attack samples significantly, more than half of them, by learning from (dis)similarity. There are 51 DoS samples classified falsely as a normal class in the case of the CE one, whereas there is only 1 sample in the case of the SupCon ResNet. Moreover, the SupCon ResNet also decreases the number of false-negative samples within the attack classes and false alarm cases. See (Fig. 9).

According to the results, the SupCon ResNet model effectively separates the classes on the Car Hacking dataset. This is because of the motivation of contrastive learning: the similar samples are pulled together while the dissimilar samples are pushed far apart. Due to the richness of the Car Hacking dataset, the SupCon ResNet can learn robust representations. These representations are useful not only for the source dataset but also for other CAN datasets, which will be proved empirically in the next section.

5.5. Transfer learning results

The SupCon ResNet produces good representations that contribute learned knowledge to train an efficient classification model using a small dataset. In this section, we will analyze the results of transfer learning from the proposed model, trained with the Car Hacking dataset, to other smaller-size and different car model datasets. We set up the experiment with three different configurations:

Random: We trained the model on new datasets from scratch. This means that the weight of the model is randomly initialized without transfer learning.
CE ResNet: We used the CE ResNet trained on the Car Hacking dataset as the pretrained model.
SupCon ResNet: We used the SupCon ResNet trained on the Car Hacking dataset as the pretrained model.

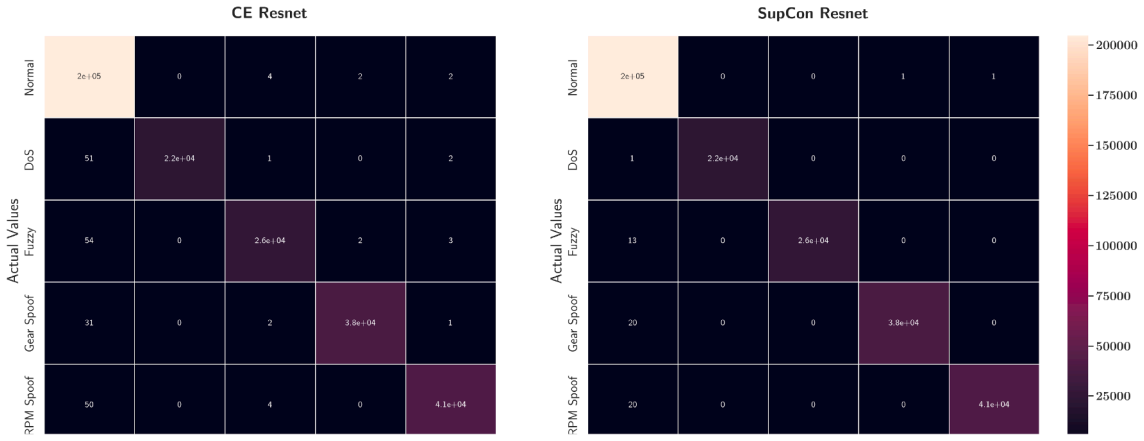

Fig. 8. Comparison of confusion matrix between CE and SupCon models.

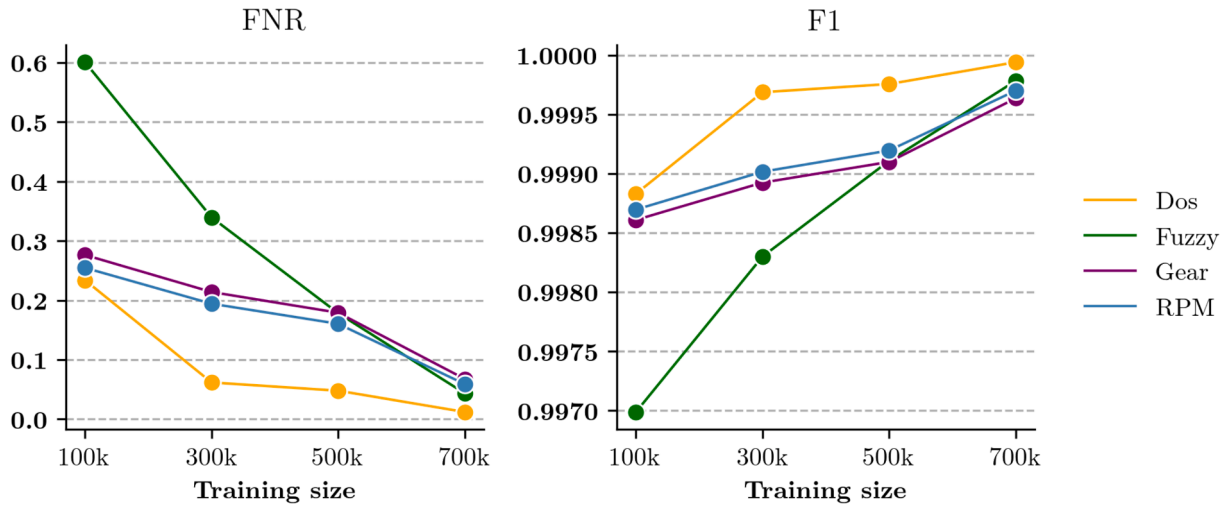

Fig. 9. The performance of the proposed model as a function of the number of training samples grouped by attack types. The metrics are FNR (lower is better) and F1 score (higher is better).

Table 5
Transfer learning results for KIA Soul data.

Attack Type	Model	FNR	Rec	Pre	F1
Normal	Random	0.04	0.9996	0.9980	0.9988
	CE ResNet	0.04	0.9996	0.9990	0.9993
	SupCon ResNet	**0.00**	**1.0**	**0.9999**	**0.9999**
DoS	Random	0.12	0.9988	**1.0**	0.9994
	CE ResNet	0.05	0.9995	**1.0**	0.9997
	SupCon ResNet	**0.01**	**0.9999**	**1.0**	**1.0**
Fuzzy	Random	0.49	0.9951	0.9994	0.9973
	CE ResNet	0.20	0.9980	0.9992	0.9986
	SupCon ResNet	**0.06**	**0.9995**	**0.9999**	**0.9997**
Malfunction	Random	0.21	0.9979	0.9987	0.9983
	CE ResNet	0.13	0.9988	0.9992	0.9989
	SupCon ResNet	**0.03**	**0.9997**	**1.0**	**0.9998**
Overall	Random	0.21	0.9979	0.9987	0.9983
	CE ResNet	0.13	0.9988	0.9992	0.9989
	SupCon ResNet	**0.03**	**0.9997**	**1.0**	**0.9998**

Table 6
Transfer learning results for Chevrolet Spark data.

Attack Type	Model	FNR	Rec	Pre	F1
Normal	Random	0.25	0.9975	0.9960	0.9967
	CE ResNet	**0.11**	**0.9989**	0.9982	0.9985
	SupCon ResNet	**0.11**	**0.9989**	**0.9992**	**0.9991**
DoS	Random	0.18	0.9982	**1.0**	0.9991
	CE ResNet	0.05	0.9995	**1.0**	**0.9998**
	SupCon ResNet	**0.04**	**0.9996**	**1.0**	**0.9998**
Fuzzy	Random	2.99	0.9701	0.9731	0.9719
	CE ResNet	1.46	0.9854	0.9901	0.9878
	SupCon ResNet	**0.47**	**0.9953**	**0.9902**	**0.9927**
Malfunction	Random	0.28	0.9972	0.9994	0.9983
	CE ResNet	0.13	0.9987	0.9989	0.9988
	SupCon ResNet	**0.04**	**0.9996**	**1.0**	**0.9998**
Overall	Random	0.93	0.9908	0.9923	0.9915
	CE ResNet	0.44	0.9956	0.9968	0.9962
	SupCon ResNet	**0.17**	**0.9984**	**0.9974**	**0.9979**

The results of KIA Soul and Chevrolet Spark models are presented in Table 5 and Table 6, respectively.

Overall, transfer learning reduces FNRs and increases the F1 score, especially for the fuzzy attack. For example, on both two target datasets, the FNRs of the fuzzy attack reduce by two times when using the CE ResNet as a pretrained model, compared to the random model. With the SupCon ResNet model, this number decreases significantly to 0.06 % for the KIA Soul and 0.47 % for the Chevrolet Spark, respectively. We can see that the dataset from Chevrolet Spark is more difficult to detect than the dataset from KIA Soul, as the models of Chevrolet Spark produce higher FNRs and lower F1 scores. This is because the training dataset of Chevrolet Spark is smaller than that of KIA Soul. For example, there are only almost 6,000 fuzzy attack messages in the Chevrolet Spark dataset, which is much fewer than 39,812 samples in the KIA Soul dataset. Even

in the case of a substantially small dataset, the model transferred from the SupCon ResNet still achieves the lowest FNR of 0.47 %. To sum up, when using the SupCon ResNet as a pretrained model, the performance of the target model improves significantly. From the experiment results, we observe two important conclusions as follows:

- When the CAN dataset is relatively small, transfer learning from a different car model can improve the detection ability in terms of FNR and F1 score.
- It is more accurate to obtain knowledge using the SupCon ResNet model than with the CE ResNet model. In fact, the SupCon ResNet increases the overall F1 score.

5.6. The effect of training size

In this subsection, we analyze model sensitivity by examining how training size affects the performance of the proposed model. We evaluated the model from two different perspectives: its performance on its own and its capability for transfer learning tasks.

From the first perspective, we adjusted the number of training samples within the range of [100 k, 700 k] and visualize the FNRs and F1 scores of these models for different attack types in Fig. 7. It is clear that as we increase the amount of training data, the prediction model performs better, resulting in lower FNRs and higher F1 scores. The results also confirm that the fuzzy attack is the most challenging, consistent with the findings in the previous section. Interestingly, the gaps in FNR and F1 scores between different attack types diminish significantly with an increase in training samples. Specifically, the gap between the FNR of fuzzy attacks and the others is approximately 0.3 % when the number of training samples is 100 k. The value decreases to less than 0.1 % when the number of training samples reaches 700 k.

From the second perspective, we applied the four models in the first setting to transfer learning for the KIA Soul and Chevrolet Spark models. Fig. 8 shows the overall FNRs of models employed from this setting. On one hand, the results for the KIA Soul dataset demonstrate the robustness of the proposed model to variations in the number of training samples. All FNR values are below 0.4 %. Notably, even in the extreme setting where the training samples of source and target datasets are 100 k and 10 k, respectively, the FNR remains as low as 0.35 %. On the other hand, the results of the Chevrolet Spark dataset reveal challenges in transfer learning. This can be explained by a large discrepancy in the data pattern of CAN signals between the source dataset (Hyundai model) and the Chevrolet Spark. However, as shown in Fig. 10, an FNR lower than 0.1 % can be achieved by a model trained from at least 20 k training samples of target datasets and using the pretrained model trained from at least 300 k training samples.

Table 7
Model Complexity Comparison.

Model	#Parameters (M)	MACs (M)
Inception ResNet (Song et al., 2020)	1.69	97.19
Rec-CNN (Desta et al., 2022)	0.53	47.71
Transformer (Nguyen et al., 2023)	2.67	240.46
Ours	0.7	32.56

5.7. Model complexity analysis

Along with high detection accuracy, a CAN-bus IDS must follow hardware constraints, such as low computing power, small memory size, and small response time, such that it can be installed and deployed in an ECU. Table 7 shows the complexity of the proposed model in terms of the number of parameters and multiply-accumulate (MAC) operations [27], compared to other CNN-based and transformer-based methods. While the number of parameters relates to the memory size of a deep learning model, the MACs involve the speed of the model. Compared to the Inception ResNet, our model is more lightweight and expected to run faster. Although the Rec-CNN has less parameters, the model generates bigger MACs due to large fully connected layers. Consequently, the Rec-CNN is expected to run slower than our proposed model. Furthermore, transformer-based method has the highest computational demands due to the intricateness of the attention mechanism, despite its outstanding performance. In particular, the number of total parameters of the transformer network is approximately four times greater than that of our model. The proposed model was deployed on the NVIDIA Jetson AGX Xavier to test the resource constraints. We chose the NVIDIA Jetson AGX Xavier, (which has 512-core Volta GPU with Tensor Cores, 8-core ARM CPU, and memory of 32 GB RAM), because it can be integrated into the vehicle's XPU to detect intruders in CAN bus [14]. The result shows that the proposed model spends approximately 5.96 ms to detect one frame containing 29 messages. This implies that the model can process up to approximately 4800 messages within a second. Simultaneously, there are approximately 2000 CAN messages on the bus (Song et al., 2020). Therefore, our proposed model has enough capability to be deployed in a real ECU.

6. Conclusion

This study aims to develop a lightweight and efficient CAN bus intrusion detection system capable of detecting and identifying specific

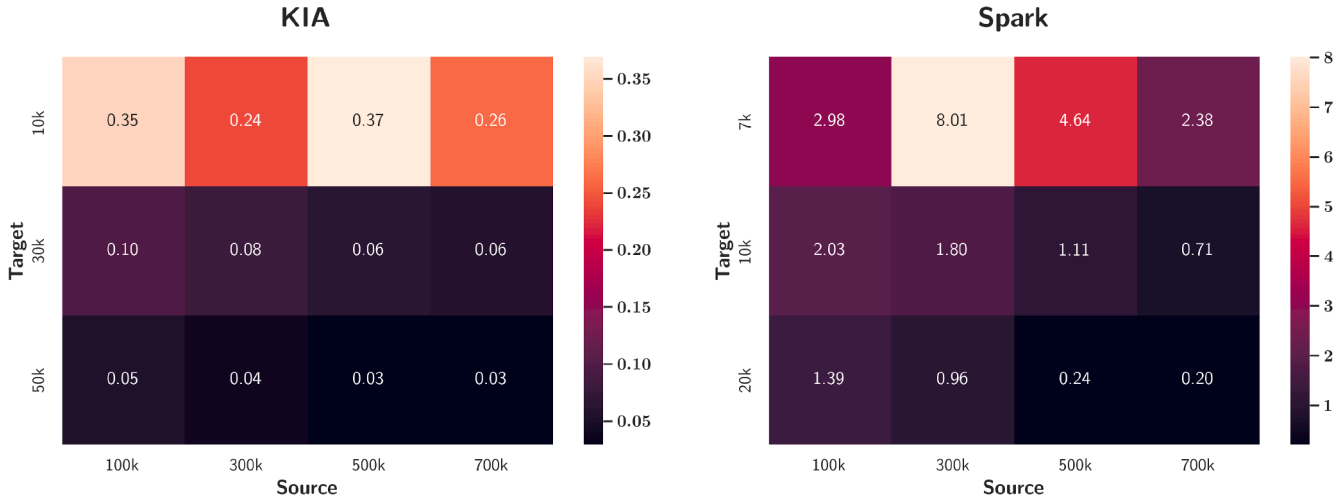

Fig. 10. The effect of training set size (of both source and target datasets) on transfer learning problem. The metric used for the illustration is FNR (lower is better).

types of attacks. We introduced a SupCon ResNet, which was trained with the supervised contrastive loss, by manipulating the CAN ID sequences in binary form. To the best of our knowledge, this is the first study that combined supervised contrastive learning and transfer learning to the in-vehicle IDS. The results of experiments on real datasets indicate that the supervised contrastive loss significantly reduced the false-negative rate, especially for the fuzzy attack. Consequently, a lightweight deep learning model, which occupies low memory size and runs faster, can be deployed in an ECU to serve the intrusion detection task without detection accuracy degradation. Moreover, the proposed model is useful for transfer learning, solving the challenge of data lacking in the new release car model. We showed that the transfer learning from the SupCon ResNet model is superior to other baselines. Concretely, the proposed model decreases the false-negative rate by four and more than two times when tested with KIA Soul and Chevrolet Spark models, respectively. Drawing upon these results, it is evident that the SupCon ResNet adeptly addresses the classification of multiple attacks on the CAN bus. Furthermore, the model saves time and labor costs for data collection for a new car model. However, the proposed system contains a limitation of no unknown attack detection. The model should be frequently updated to detect new types of attacks, since the multiclass classification cannot be solved using unsupervised models. The resolution of this challenge could involve the application of online retrainable neural network techniques (A. D. Doulamis, Doulamis, & Kollias, 2000), which presents a promising avenue for future exploration.

CRediT authorship contribution statement

Thien-Nu Hoang: Conceptualization, Methodology, Software, Visualization, Writing – original draft. **Daehee Kim:** Validation, Conceptualization, Formal analysis, Supervision, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1A4A2001810, 2022H1D8A3038040, RS-2023-00277255), by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-01197, Convergence security core talent training business (SoonChunHyang University)), and this work was supported by the Soonchunhyang Research Fund.

References

Ahmed, I., Ahmad, A., & Jeon, G. (2021). Deep learning-based intrusion detection system for internet of vehicles. *IEEE Consumer Electronics Magazine*. https://doi.org/10.1109/MCE.2021.3139170

Andresini, G., Appice, A., & Malerba, D. (2021). Autoencoder-based deep metric learning for network intrusion detection. *Information Sciences, 569*, 706–727. https://doi.org/10.1016/J.INS.2021.05.016

Ashraf, J., Bakhshi, A. D., 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, 1–12*. https://doi.org/10.1109/tits.2020.3017882

Avatefipour, O., Saad Al-Sumaiti, A., El-Sherbeeny, A. M., Mahrous Awwad, E., Elmeligy, M. A., Mohamed, M. A., & Malik, H. (2019). An intelligent secured framework for cyberattack detection in electric vehicles' can bus using machine learning. *IEEE Access, 7*, 127580–127592. https://doi.org/10.1109/ACCESS.2019.2937576

BOSCH CAN Specification Version 2.0. (1991).

Chen, K., Rögnvaldsson, T., Nowaczyk, S., Pashami, S., Johansson, E., & Sternelöv, G. (2022). Semi-supervised learning for forklift activity recognition from controller area network (CAN) signals. Sensors 2022, Vol. 22, Page 4170, 22(11), 4170. https://doi.org/10.3390/S22114170.

Chen, M., Zhao, Q., Jiang, Z., & Xu, R. (2021). Intrusion detection for in-vehicle CAN networks based on auxiliary classifier GANs. 2021 International Conference on High Performance Big Data and Intelligent Systems, HPBD and IS 2021, 186–191. https://doi.org/10.1109/HPBDIS53214.2021.9658465.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. 37th International Conference on Machine Learning, ICML 2020, PartF168147-3, 1575–1585. https://doi.org/10.48550/arxiv.2002.05709.

Cho, K.-T., & Shin, K. G. (2016). *Fingerprinting Electronic Control Units for Vehicle Intrusion Detection., 911*. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/cho.

Chopra, S., Hadsell, R., & Lecun, Y. (2005). Learning a Similarity Metric Discriminatively, with Application to Face Verification. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*.

D'Angelo, G., Castiglione, A., & Palmieri, F. (2020). A cluster-based multidimensional approach for detecting attacks on connected vehicles. *IEEE Internet of Things Journal*. https://doi.org/10.1109/JIOT.2020.3032935

Derhab, A., Belaoued, M., Mohiuddin, I., Kurniawan, F., & Khan, M. K. (2021). Histogram-based intrusion detection and filtering framework for secure and safe in-vehicle networks. *IEEE Transactions on Intelligent Transportation Systems, 1–14*. https://doi.org/10.1109/TITS.2021.3088998

Desta, A. K., Ohira, S., Arai, I., & Fujikawa, K. (2022). Rec-CNN: In-vehicle networks intrusion detection using convolutional neural networks trained on recurrence plots. *Vehicular Communications, 35*, Article 100470. https://doi.org/10.1016/J.VEHCOM.2022.100470

Doulamis, A. D., Doulamis, N. D., & Kollias, S. D. (2000). On-line retrainable neural networks: Improving the performance of neural networks in image analysis problems. *IEEE Transactions on Neural Networks, 11*(1), 137–155. https://doi.org/10.1109/72.822517

Doulamis, N., & Doulamis, A. (2014). Semi-supervised deep learning for object tracking and classification. In *2014 IEEE International Conference on Image Processing*. https://doi.org/10.1109/ICIP.2014.7025170

Han, M. L., Kwak, B. I., & Kim, H. K. (2018). Anomaly intrusion detection method for vehicular networks based on survival analysis. *Vehicular Communications, 14*, 52–63. https://doi.org/10.1016/J.VEHCOM.2018.09.004

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-December, 770–778. https://doi.org/10.48550/arxiv.1512.03385.

Hoang, T.-N., & Kim, D. (2022). Detecting in-vehicle intrusion via semi-supervised learning-based convolutional adversarial autoencoders. *Vehicular Communications, 38*, Article 100520. https://doi.org/10.1016/J.VEHCOM.2022.100520

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

Jo, H. J., & Choi, W. (2021). A survey of attacks on controller area networks and corresponding countermeasures. *IEEE Transactions on Intelligent Transportation Systems*. https://doi.org/10.1109/TITS.2021.3078740

Kang, L., & Shen, H. (2021). A Transfer Learning based Abnormal CAN Bus Message Detection System. In *Proceedings - 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems*. https://doi.org/10.1109/MASS52906.2021.00073

Kang, M. J., & Kang, J. W. (2016). A novel intrusion detection method using deep neural network for in-vehicle network security. IEEE Vehicular Technology Conference, 2016-July. https://doi.org/10.1109/VTCSPRING.2016.7504089.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Research, G., Tian, Y., Isola, P., Maschinot, A., Liu, C., & Krishnan, D. (2020). Supervised Contrastive Learning. https://doi.org/10.48550/arxiv.2004.11362.

Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., … Savage, S. (2010). Experimental Security Analysis of a Modern Automobile. In *2010 IEEE Symposium on Security and Privacy* (pp. 447–462).

Liu, Q., Wang, D., Jia, Y., Luo, S., & Wang, C. (2022). A multi-task based deep learning approach for intrusion detection. *Knowledge-Based Systems, 238*, Article 107852. https://doi.org/10.1016/J.KNOSYS.2021.107852

Lopez-Martin, M., Sanchez-Esguevillas, A., Arribas, J. I., & Carro, B. (2022). Supervised contrastive learning over prototype-label embeddings for network intrusion detection. *Information Fusion, 79*, 200–228. https://doi.org/10.1016/J.INFFUS.2021.09.014

Loshchilov, I., & Hutter, F. (2016). SGDR: Stochastic Gradient Descent with Warm Restarts. 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings. https://doi.org/10.48550/arxiv.1608.03983.

Mammeri, A., Zhao, Y., Boukerche, A., Siddiqui, A. J., & Pekilis, B. (2019). Design of a semi-supervised learning strategy based on convolutional neural network for vehicle maneuver classification. In *7th IEEE International Conference on Wireless for Space and Extreme Environments*. https://doi.org/10.1109/WISEE.2019.8920301

Nam, M., Park, S., & Kim, D. S. (2021). Intrusion detection method using bi-directional GPT for in-vehicle controller area networks. *IEEE Access, 9*, 124931–124944. https://doi.org/10.1109/ACCESS.2021.3110524

Nguyen, T. P., Nam, H., & Kim, D. (2023). Transformer-based attention network for in-vehicle intrusion detection. *IEEE Access, 11*, 55389–55403. https://doi.org/10.1109/ACCESS.2023.3282110

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering, 22*(10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

Seo, E., Song, H. M., & Kim, H. K. (2018). GIDS: GAN based intrusion detection system for in-vehicle network. 2018 16th Annual Conference on Privacy, Security and Trust, PST 2018. https://doi.org/10.1109/PST.2018.8514157.

Song, H. M., Woo, J., & Kim, H. K. (2020). In-vehicle network intrusion detection using deep convolutional neural network. *Vehicular Communications, 21*, Article 100198. https://doi.org/10.1016/j.vehcom.2019.100198

Sun, H., Chen, M., Weng, J., Liu, Z., & Geng, G. (2021). Anomaly detection for in-vehicle network using CNN-LSTM with attention mechanism. *IEEE Transactions on Vehicular Technology, 70*(10), 10880–10893. https://doi.org/10.1109/TVT.2021.3106940

Taylor, A., Leblanc, S., & Japkowicz, N. (2016). Anomaly detection in automobile control network data with long short-term memory networks. In *Proceedings - 3rd IEEE International Conference on Data Science and Advanced Analytics*. https://doi.org/10.1109/DSAA.2016.20

Zhu, X. (Jerry). (2005). Semi-supervised learning literature survey. https://minds.wisconsin.edu/handle/1793/60444.