2021 IEEE Intelligent Transportation Systems Conference (ITSC)
Indianapolis, USA. September 19-21, 2021

In-Vehicle Network Attack Detection Across Vehicle Models: A Supervised-Unsupervised Hybrid Approach

Shu Nakamura†, Koh Takeuchi†, Hisashi Kashima†,
Takeshi Kishikawa§, Takashi Ushio§, Tomoyuki Haga§, Takamitsu Sasaki§
†Kyoto University, §Panasonic Corporation

2021 IEEE International Intelligent Transportation Systems Conference (ITSC) | 978-1-7281-9142-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ITSC48978.2021.9564896

***Abstract*— Recent studies have demonstrated that the injection of malicious messages into in-vehicle networks can cause unintended operation of the controls of vehicles, which has been highlighted as one of the most serious and urgent issues that threaten the safety of automobiles. Attempts have been made to use supervised and unsupervised machine learning for automatic, data-driven intrusion detection. However, previous approaches considered only the detection and classification of attacks on a target car based on the data of the same model of car; they are relatively ineffective when the objective is to handle new car models for which not many data are yet available. In this paper, we address the task of detecting and classifying malicious messages injected into in-vehicle networks by transferring "knowledge" from different car models for which ample data exist. In our proposed approach to the dataset of new car models a pretrained classification model that was supervised-trained on the data of previous car models is combined it with an unsupervised detector that uses only the normal data from the target car. The advantage of the proposed approach is that it does not require past data and therefore is applicable to various scenarios. The results of our experiments using in-vehicle CAN messages datasets collected from three different cars show the effectiveness of the proposed approach.**

I. Introduction

In recent automobile models, multiple computing units called electronic control units (ECUs) communicate with each other via a controller area network (CAN) to control the operation of their components, such as the brakes, steering, and throttle. Furthermore, these systems are frequently connected to external systems via 3G/4G networks and Bluetooth, which increases their security vulnerability. In addition, a CAN has no security features to protect it from external malicious attacks, which is considered a serious vulnerability of the current CAN. Recent studies have shown that this is not fiction, but that malicious attackers can indeed control the operation of a vehicle by injecting packets into the CAN. In 2010, Kascher *et al.* demonstrated that malicious CAN packets can in fact realize control operations, such as locking the brakes or stopping the engine while the car is being driven, from the outside the vehicle [1]. Miller also showed that a car's steering and braking can be controlled by packets injected into the CAN [2]. In 2015, the Chrysler Corporation announced that it would recall 1.4 million vehicles and refund their purchasers because of possible vulnerabilities.

In response to the above issues, attempts have recently been made to detect automatically malicious attacks from outside a car in a data-driven manner. Because rule-based methods designed by human experts cannot handle the ever-evolving attack methods, data-driven machine learning approaches are being studied. Two types of machine learning-based approaches exist: supervised and unsupervised. The former requires annotated data for each attack type, but can classify attack types in detail. Conversely, the latter cannot classify attack types, but does not require annotations and can be expected to detect unknown types of attack. Thus, each method has its own strengths and weaknesses, but the challenge that all these approaches face is the same: to provide CAN security support for new car models. In the supervised approach, there is no guarantee that the detection rules for previous vehicle models can be applied to a new target vehicle. Although the CAN specifications are standardized, the assignment of IDs, the timing of message transmission, and the grammar of the message content are different for each car model. An additional problem is that training data collected for other car models are not always available. This may be because, for example, only an attack detection model trained by a third party is provided, from the perspective of protecting intellectual property or confidential information, or simply because those who produce these data wish to avoid retraining a model using a large amount of data. Furthermore, the data for the target car model is not plentiful, although normal data can be collected relatively easily, and the unsupervised learning approach cannot classify attacks.

In this paper, we propose a hybrid approach that performs attack detection and classification for a target car using both (1) a supervised attack detection and classifier model pretrained on the historical data of other car models (termed here source car models) and (2) the normal data of the new target car. In the proposed approach, the predictions of the pretrained classifier are compensated by the results of an unsupervised detector trained on the normal data of the target car, which is implemented by using an auto-encoder detector based on reconstruction errors. The proposed method appropriately transfers the knowledge gained from previous

source vehicles to the new target vehicle, enabling attack detection and classification.

In our experiments using real data, the detector that was pretrained only on the data of the source car model was used as the baseline, and the detector that was retrained on the data of all the source and target car models was used as the reference model, which provides the upper bound on the performance. The performance of the proposed approach was compared with that of these two methods. The results show that, although its performance does not reach that of the retrained model, which is the upper bound, the proposed method outperforms the baseline method, indicating that it is effective.

II. Related Work

Even before the introduction of CAN in automobiles, data-driven detection of malicious attacks on various computer networks had been studied for many years. To detect injection attacks such as flooding attacks, various metrics, such as moving averages, cumulative sums (CUSUM), and spectral analysis, were utilized. Huang *et al.* [3] proposed an adaptive version of CUSUM, and Bhatia *et al.* [4] proposed a method that detects attacks on the modbus flooding system by setting a threshold using the exponentially weighted moving average (EWMA) in the time interval of the packet. Many studies have addressed distributed denial-of-service (DDoS) attacks [5]–[7]. Rahmani *et al.* [5] and Li *et al.* [6] proposed a method for detecting DDoS attacks using f-divergence and entropy, respectively.

Detection methods specific to the detection of attacks on CAN can be divided into payload-based approaches using the content of the message and time-stamp-based approaches based on the arrival time and interval of the message.

The methods in the former category are focused on the contents of the transmitted packets and detect abnormal packets whose contents differ significantly from the contents of normal messages. The method proposed by Müter and Asaj used the entropy of packet contents [8], and Marchetti *et al.* demonstrated the effectiveness of their entropy-based method using real-world CAN traffic data collected from unmodified licensed vehicles [9]. Kang and Kang [10] formulated the attack detection problem as a supervised classification problem and trained a deep neural network using both malicious and normal packets. Taylor *et al.* [11] used the concept of prediction-based anomaly detection; they trained a long short-term memory (LSTM) neural network to predict the next packet, and, if its actual content was significantly different from the predicted content, they considered the packet malicious. Tanaka *et al.* [12] proposed a method for unsupervised attack detection using change detection techniques, as well as a method for indicating the basis for the decisions.

In time stamp-based detection methods, a packet is considered anomalous when the time interval between two consecutive messages deviates significantly from that of normal ones. They are based on the assumption that most ordinary packets are sent periodically [13]. Hamada *et al.* modeled normal intervals using a Gaussian mixture model [14]. Taylor *et al.* [15] and Kuwahara *et al.* [16] proposed unsupervised learning methods for detecting malicious packet sequences.

As summarized above, various developments in data-driven attack detection methods using machine learning have occurred, but no attempts at domain adaptation [17] to transfer knowledge across multiple car models have been made.

III. Problem Settings: Transferring Past Experiences to a New Car Model

Our objective in this study is to monitor messages sent over a CAN in a target car model, such as a newly launched one, detect malicious attack messages injected from outside, and classify them into attack types.

More formally, suppose that there is a set $A = \{a_0, a_1, \ldots, a_C\}$ of predetermined C-type attacks, where it is assumed for convenience that a_0 is the attack-free state. The objective is to construct an attack detector that can classify each message $m \in M^{\text{test}}$ of the set of messages $M^{\text{test}} = \{m_1^{\text{test}}, m_2^{\text{test}}, \ldots\}$ of the target car model as into one of set A. To achieve this goal, we assume that (1) an attack detector for the other vehicle types (termed the source vehicle types) and (2) a set of CAN messages when the target vehicle type is not under attack are available.

Our problem setting assumes the following scenario, which is likely to occur in real-world operations. Assume that there exist sufficient data for our purpose pertaining to previously released source car models, for both various types of attacks and non-attacks, collected over a relatively long period. Using these data, we can build an attack detector (by supervised learning) for the source car models. However, we assume that the data for the source car models are not accessible, i.e., it is not even possible to obtain them from a third party, and that only the attack detector is available. Later, in our experiments we will compare this case with that where the source model data are also available. However, it is also natural to assume that for a (newly released) target vehicle model, data under normal (attack-free) driving conditions can be obtained relatively easily. However, it is usually difficult to obtain data on actual attacks or simulated attacks.

We now present a formal problem setting of attack detection in a new vehicle model in which the experiences of past vehicle models are transferred to the dataset of a new model. Suppose we have S car models, of which $S-1$ are the source car models and the remaining model is the target car model. A sufficient amount of data consisting

of normal messages and C types of attack messages exist for the source cars, and an attack detector $f^{\text{source}} : M \to \mathbb{R}_+^{C+1}$ trained on these messages is already available. Here, M is the feature vector space of messages that f^{source} receives as its input, and is constructed based on the time of message transmission, CAN ID, and message contents, as described in the experiments section. Note that, for convenience, we equate a message m with its feature representation m and use them interchangeably. The output of f^{source} is a vector representing the degree to which the input message belongs to normal and C-type attack messages. For a message $m \in M$, the first dimension of $f^{\text{source}}(m)$ represents the likelihood that m is a normal message and the c+1-th dimension represents the likelihood that m is a message of attack type a_c.

We assume that f^{source} is provided by a third party, or that we are not allowed direct access to historical data and the trained parameters of f^{source} for privacy reasons, or that we wish to avoid the cost of retraining the model with large amounts of historical data. However, we assume that, for the target car model, only the normal data and no attack data are available. Let $M^{\text{train}} = \{m_1, m_2, \ldots m_N\}$ be the N attack-free messages.

The problem setting we consider in this study is summarized as:

- **INPUT:** a pretrained attack detector $f^{\text{source}} : M \to \mathbb{R}_+^{C+1}$ trained on the source car data and a set of attack-free data M^{train} collected for a target car.
- **OUTPUT:** an attack detector $f^{\text{target}} : M \to \mathbb{R}_+^{C+1}$ for the target car model.

IV. Proposed Method: A Hybrid Detection Approach using Both Supervised and Unsupervised Detectors

In this paper, we propose a hybrid approach that performs attack detection and classification for a target car using both (1) $f^{\text{source}} : M \to \mathbb{R}_+^{C+1}$, a supervised attack detection and classifier pretrained from the historical data of source car models, and (2) $M^{\text{train}} = \{m_1, m_2, \ldots m_N\}$, the normal data of the new target car.

Although the specifications of CAN are standardized, the assignment of IDs, the timing of message transmission, and the detailed grammar of the message content vary among different car models. Therefore, it is not always possible to apply knowledge gained from the source car models to the target car model. Ideally, it would be desirable to train a new detector using all the data from the source and target car models. However, in our setup, the data from the source car model are not available. Our concept is that the predictions of the pretrained classifier are corrected by an unsupervised detector trained on the normal data of the target car.

Suppose an unsupervised anomaly detector provides the anomaly probability $a(x) \in [0, 1]$ for message m. (The construction of this detector is described later.) In addition, as mentioned previously, the pretrained classifier f^{source} provides how likely m is a normal message, $f_0^{\text{target}}(m)$. It also provides $f_c^{\text{source}}(m)$, that is, gives how likely m is an attack message of type $c \in \{1, 2, \ldots, C\}$. By combining these two models, the final detector/classifier scores are provided by f^{target} as

$$f_0^{\text{target}}(m) = f_0^{\text{source}}(m)\,(1 - a(m)) \quad (1)$$

$$f_c^{\text{target}}(m) = f_c^{\text{source}}(m) \cdot a(m) \text{ for } c = 1, 2, \ldots, C. \quad (2)$$

Eq. (1) represents the score of the pretrained classifier that judges the message to be normal weighted by the probability that the unsupervised detector judges it to be normal, whereas Eq. (2) weights the c-th attack score given by the pretrained classifier by the probability that the unsupervised detector determines this message to be an attack. The above can be interpreted to mean that the decision of the pretrained classifier is corrected by the unsupervised detector of the target car model. Note that the unsupervised detector can detect only the presence of an attack, but not its type.

We realized the unsupervised detector by using an auto-encoder detector based on reconstruction errors. An auto-encoder is a type of multilayer neural network that performs unsupervised learning and consists of two serially connected networks, encoder h and decoder g. The encoder network h takes the message feature vector m as the input and obtains the latent representation $h(m)$, and the decoder network g returns the latent representation h to the original message space. The auto-encoder learns h and g to make the decoded message as close as possible to the original message m in terms of the reconstruction error

$$\sum_{m \in M^{\text{train}}} \|m - g(h(m))\|^2.$$

In the detection phase (test phase), we use the reconstruction error of each message m in the test data $M^{\text{test}} = \{m_1^{\text{test}}, m_2^{\text{test}}, \ldots\}$ of the target car model as the anomaly score $r(m)$; that is,

$$r(m) = \|m - g(h(m))\|^2.$$

From this definition, the anomaly score is an unbounded non-negative value, and therefore, we convert it to

$$a(m) = 1 - \exp(\lambda r(m))$$

to utilize it as the "anomaly probability." Here, λ is a scaling parameter; we determine λ such that the anomaly score for the attack-free data does not exceed a certain threshold τ, that is,

$$\max_{m \in M^{\text{train}}} a(m) = \tau.$$

In our experiments, we specifically set $\tau = 0.05$.

The proposed method appropriately transfers the knowledge gained from previous source vehicles to the dataset of the new target vehicle, enabling attack detection and attack classification.

TABLE I: Size of the training datasets. Each number indicates the number of messages of a particular attack type. 'Atk-free' represents attack-free messages, and 'Malfunc.' represents malfunction attacks.

Target car model	Type of attack			
	Atk-free	Flooding	Fuzzy	Malfunc.
KIA	473212	120000	460000	400530
HYUNDAI	431135	96000	403299	419689
CHEVROLET	136934	85000	41000	51000

V. Experiments

A. Experimental Settings

1) Datasets: We conducted our experiments on a benchmark dataset that includes three different attacks on three different car models produced by three different manufacturers, which was made public by Han *et al.* [18][1]. These data contain CAN message data communicated to three car models: KIA Soul, Hyundai YF Sonata, and Chevrolet Spark; the three types of attacks ($C = 3$) are as follows:

1) Flooding: all the CAN IDs and data contents of the attack messages have been replaced with 0.
2) Fuzzy: all the CAN IDs and data contents of the attack messages have been replaced with random numbers.
3) Malfunction: Several types of data, specific to each car model, that cause the car to behave incorrectly have been sent to the CAN ID representing the device being attacked.

Each CAN message has been assigned one of four labels, representing the three attack types and the attack-free state. The size of the dataset is summarized in Table I.

From the information contained in each CAN message, we used the time stamp, CAN ID, and message content in our experiments and processed these data to construct a feature vector for each message. For the time stamp, we followed the time-stamp-based approach and used the time interval from the previous message that had the same CAN ID as one of the features. Because this is the only real-valued feature that takes a relatively small value, we scaled it such that it had mean 0 and variance 1 in order to scale with the other features.

Because the CAN ID was between 0 and 2047, it was expressed in 11 bits and used as 11-dimensional binary features. We assumed no prior knowledge of the data format and used the bit strings as they were. The amount of the 192-bit message area that was in fact used depended on each message, and in many cases the latter part of the message area was not used. Therefore, we represented an unused bit by `NaN`. In other words, each bit took one of the values $\{1, 0, \texttt{NaN}\}$. This means that the message content part was represented as a 192-dimensional ternary feature vector. Ultimately, each message was represented as a 204-dimensional vector.

[1]Available for academic purposes at `https://ocslab.hksecurity.net/Datasets/survival-ids`.

2) Comparison methods: The proposed hybrid model was compared with the pretrained model as the baseline and the retrained model, which is an ideal model that provided the performance upper bound.

Pretrained model (baseline): The pretrained model was learned only from the training data of the source car models and was then directly applied to the test data of the target car model for evaluation. LightGBM [19] was used as the supervised classification model.

Retrained model (ideal performance upper bound): In the retrained model, in addition to the training data of the source car models, the attack-free training data of the target car model were also used to train the prediction model. In our problem setup, the training data of the source car model were not available for training, and only the pretrained model could be used. Therefore, the retrained model was used as a reference record, as it gives the upper limit of the ideal performance.

Hybrid model (proposed): In the proposed hybrid model, an anomaly detector using an auto-encoder was trained on the attack-free training data of the target car model. This was combined with the pretrained model described above.

3) Implementation details: The pretrained model, which was trained on the data of the source car model, and the retrained model, which provided the ideal performance upper bound, were implemented using LightGBM [19] with the default settings. The auto-encoder used as the unsupervised anomaly detector in the proposed method was implemented using PyTorch. It is a three-layer fully connected neural network with one hidden layer. The dimension of the hidden layer was 10, and the sigmoid function was used as the activation function. The objective function was the squared reconstruction error. The learning rate was set to 0.1 and all the other parameters were set to their default values.

4) Evaluation settings: Two of the three car models ($S = 3$) were used as the source car models and the remaining model as the target car model. By varying the target car model, the evaluation was performed three times for three different settings.

For training the pretrained model, the attack detector/classifier was trained by a supervised multi-class classification method using the mixed attack-free and attack data of the source car model. In the case of the retrained model, the attack detector was trained by adding the attack-free data of the target car model to the training data as the attack-free class. In the case of the proposed hybrid model, in addition to the pretrained model, an unsupervised detector (auto-encoder) was trained using the attack-free data of the target car model.

In the evaluation, the mixed attack/attack-free data of the target car model were used. Separate datasets were prepared for each of the three types of attack, and these were mixed together. In the evaluation, we used the confusion matrix to check for errors in attack types,

as well as the precision and recall scores for each attack. In addition, we compared the accuracy and the F1 value of the different methods to evaluate the overall accuracy.

B. Results

Tables II, III, and IV show the confusion matrices for the cases where we took each one of the three car models (KIA, HYUNDAI, CHEVROLET) as the target car model, respectively. The tables also show the precision and recall scores in the bottom and the rightmost rows, respectively. The numbers of correct predictions are shown on the diagonal of the confusion matrices. The off-diagonal entries are the number of incorrect predictions; in particular, the off-diagonal entries in the Atk-free row are false positives, wheres those entries of the Atk-free column in other rows correspond to false negatives.

The tables show that, in most cases, the proposed hybrid model outperformed the baseline pretrained model, and its performance was sometimes comparable to that of the retrained model, which was considered to provide the performance upper bound in our scenario. The baseline pretrained model suffered from false positive predictions, especially in fuzzy attacks; this is because a fuzzy attack injects random messages, and therefore, the normal behavior of the target car model should be captured, rather than the attack behavior. In contrast to the baseline model, the proposed model successfully reduced the number of false positive predictions in exchange for a relatively small number of false negatives; this is probably because the proposed model captures the behavior of the target car model under attack-free conditions and filters the false positives appropriately.

The precision and recall values were also constantly improved by the hybrid method and were sometimes comparable to those of the retrained model. The only exception was in the case of the malfunction attacks in the CHEVROLET model; the use of normal data for the target car model hindered the detection by making the decisions overly conservative. The reduction of this type of "over-adaptation" will be a future challenge. The performance for fuzzy attacks is poor, in general. Taken together with the phenomena described above, the behavior of the CHEVROLET data may be considerably different from that of the other car two models.

Finally, Table V and Table VI show the overall F1-scores and accuracy values, respectively. It can be seen that, overall, the proposed method shows a significant improvement over the baseline pretrained model in terms of both detection and classification performance scores.

TABLE II: Confusion matrix for the KIA Soul dataset of (a) the baseline pretrained model, (b) the proposed hybrid model, and (c) the ideal retrained model. The precision and recall scores are also shown.

(a) Pretrained (baseline)

True atk type	Predicted attack type				
	Atk-free	Flooding	Fuzzy	Malfunc.	Recall
Atk-free	707200	198	188732	0	0.789
Flooding	1	16071	0	0	1.000
Fuzzy	114	1	119436	0	0.999
Malfunc.	0	0	0	67080	1.000
Precision	1.000	0.988	0.388	1.000	

(b) Hybrid (proposed)

True atk type	Predicted attack type				
	Atk-free	Flooding	Fuzzy	Malfunc.	Recall
Atk-free	887528	0	8602	0	0.990
Flooding	40	16032	0	0	0.998
Fuzzy	4627	0	114924	0	0.961
Malfunc.	13	0	0	67067	1.000
Precision	0.995	1.000	0.930	1.000	

(c) Retrained (ideal)

True atk type	Predicted attack type				
	Atk-free	Flooding	Fuzzy	Malfunc.	Recall
Atk-free	884671	0	11459	0	0.987
Flooding	1	16071	0	0	1.000
Fuzzy	217	0	119334	0	0.998
Malfunc.	0	0	0	67080	1.000
Precision	1.000	1.000	0.912	1.000	

VI. Conclusion

In this paper, we addressed the task of detecting and classifying malicious messages injected into in-vehicle networks by transferring "knowledge" from different car models with ample data to a new car model. To achieve this goal, we proposed a supervised-unsupervised hybrid approach that combines a supervised attack detector/classifier trained on the data of past car models and an unsupervised anomaly detector trained on the data of an attack-free target car model. The results of our experiments using in-vehicle CAN message datasets collected from three different car models showed the effectiveness of the proposed approach.

The problem of detecting attacks across multiple vehicle models was addressed for the first time in this study, and in our opinion this is the first step toward further advancing the artificial intelligence of automotive security. As has occurred in computer network security, advances in defense technology and attack technology for CAN security will stimulate each other and they evolve together. The consideration of data-driven security technologies over a longer period and across multiple vehicle types is an important direction for our research in the future.

References

[1] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, "Experimental security analysis of a modern automobile," in Proceedings of the 31st IEEE Symposium on Security and Privacy (S&P), 2010, pp. 447–462.

[2] C. Miller and C. Valasek, "Adventures in automotive networks and control units," DEF CON, vol. 21, no. 260-264, pp. 15–31, 2013.

[3] Y. Huang, H. Li, K. A. Campbell, and Z. Han, "Defending false data injection attack on smart grid network using adaptive CUSUM test," in Proceedings of the 45th Annual Conference on Information Sciences and Systems (CISS), 2011, pp. 1–6.

[4] S. Bhatia, N. Kush, C. Djamaludin, J. Akande, and E. Foo, "Practical modbus flooding attack and detection," in Proceedings of the 12th Australasian Information Security Conference (AISC), 2014, pp. 57–65.

TABLE III: Confusion matrix for the HYUNDAI YF Sonata dataset of (a) the baseline pretrained model, (b) the proposed hybrid model, and (c) the ideal retrained model. The precision and recall scores are also shown.

(a) Pretrained (baseline)

True atk type	Predicted attack type				
	Atk-free	Flooding	Fuzzy	Malfunc.	Recall
Atk-free	744101	304	109436	0	0.871
Flooding	1	17092	0	0	1.000
Fuzzy	27	6	84611	0	1.000
Malfunc.	0	0	0	71194	1.000
Precision	1.000	0.982	0.436	1.000	

(b) Hybrid (proposed)

True atk type	Predicted attack type				
	Atk-free	Flooding	Fuzzy	Malfunc.	Recall
Atk-free	832810	144	20887	0	0.975
Flooding	113	16980	0	0	0.993
Fuzzy	3397	6	81241	0	0.960
Malfunc.	10	0	0	71184	1.000
Precision	0.996	0.991	0.795	1.000	

(c) Retrained (ideal)

True atk type	Predicted attack type				
	Atk-free	Flooding	Fuzzy	Malfunc.	Recall
Atk-free	847387	274	6180	0	0.992
Flooding	0	17093	0	0	1.000
Fuzzy	57	0	84587	0	0.999
Malfunc.	0	0	0	71194	1.000
Precision	1.000	0.984	0.932	1.000	

TABLE IV: Confusion matrix for the CHEVROLET Spark dataset of (a) the baseline pretrained model, (b) the proposed hybrid model, and (c) the ideal retrained model. The precision and recall scores are also shown.

(a) Pretrained (baseline)

True atk type	Predicted attack type				
	Atk-free	Flooding	Fuzzy	Malfunc.	Recall
Atk-free	106381	4	82812	0	0.562
Flooding	0	14999	0	0	1.000
Fuzzy	1	47	2995	0	0.984
Malfunc.	0	0	0	3995	1.000
Precision	1.000	0.997	0.035	1.000	

(b) Hybrid (proposed)

True atk type	Predicted attack type				
	Atk-free	Flooding	Fuzzy	Malfunc.	Recall
Atk-free	153738	0	35459	0	0.813
Flooding	0	14999	0	0	1.000
Fuzzy	33	47	2963	0	0.974
Malfunc.	3994	0	0	1	0.000
Precision	0.974	0.997	0.077	1.000	

(c) Retrained (ideal)

True atk type	Predicted attack type				
	Atk-free	Flooding	Fuzzy	Malfunc.	Recall
Atk-free	179662	4	9531	0	0.950
Flooding	0	14999	0	0	1.000
Fuzzy	65	51	2927	0	0.962
Malfunc.	3995	0	0	0	0.000
Precision	0.978	0.996	0.235	0	

TABLE V: Overall performance: F1 score

Target car	Pretrained	Hybrid (proposed)	Retrained
KIA	0.834	0.987	0.992
HYUNDAI	0.895	0.930	0.996
CHEVROLET	0.612	0.976	0.969

TABLE VI: Overall performance: accuracy

Target car	Pretrained	Hybrid (proposed)	Retrained
KIA	0.828	0.988	0.989
HYUNDAI	0.893	0.976	0.994
CHEVROLET	0.608	0.813	0.935

[5] H. Rahmani, N. Sahli, and F. Kamoun, "DDoS flooding attack detection scheme based on F-divergence," Computer Communications, vol. 35, no. 11, pp. 1380 – 1391, 2012.

[6] L. Li, J. Zhou, and N. Xiao, "DDoS attack detection algorithms based on entropy computing," in Proceedings of the 9th International Conference on Information and Communications Security (ICICS), 2007, pp. 452–466.

[7] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in Proceedings of the 35th IEEE Local Computer Network Conference (LCN), 2010, pp. 408–415.

[8] M. Müter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," in Proceedings of the 22nd IEEE Intelligent Vehicles Symposium (IV), 2011, pp. 1110–1115.

[9] M. Marchetti, D. Stabili, A. Guido, and M. Colajanni, "Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms," in Proceedings of the Second IEEE International Forum on Research and Technologies for Society and Industry (RTSI), 2016, pp. 1–6.

[10] M. Kang and J. Kang, "A novel intrusion detection method using deep neural network for in-vehicle network security," in Proceedings of the 83rd IEEE Vehicular Technology Conference (VTC Spring), 2016, pp. 1–5.

[11] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in Proceedings of the 3rd IEEE International Conference on Data Science and Advanced Analytics (DSAA), 2016, pp. 130–139.

[12] D. Tanaka, M. Yamada, H. Kashima, T. Kishikawa, T. Haga, and T. Sasaki, "In-vehicle network intrusion detection and explanation using density ratio estimation," in Proceedings of IEEE Intelligent Transportation Systems Conference (ITSC), 2019, pp. 2238–2243.

[13] K.-T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection," in Proceedings of the 25th USENIX Security Symposium (USENIX Security), 2016, pp. 911–927.

[14] Y. Hamada, M. Inoue, S. Horihata, and A. Kamemura, "Intrusion detection by density estimation of reception cycle periods for in-vehicle networks: A proposal," in Proceedings of the 14th International Conference on Embedded Security in Cars (ESCAR), 2016.

[15] A. Taylor, N. Japkowicz, and S. Leblanc, "Frequency-based anomaly detection for the automotive CAN bus," in Proceedings of the World Congress on Industrial Control Systems Security (WCICSS), 2015, pp. 45–49.

[16] T. Kuwahara, Y. Baba, H. Kashima, T. Kishikawa, J. Tsurumi, T. Haga, Y. Ujiie, T. Sasaki, and H. Matsushima, "Supervised and unsupervised intrusion detection based on CAN message frequencies for in-vehicle network," Journal of Information Processing, vol. 26, pp. 306–313, 2018.

[17] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345–1359, 2009.

[18] M. L. Han, B. I. Kwak, and H. K. Kim, "Anomaly intrusion detection method for vehicular networks based on survival analysis," Vehicular Communications, vol. 14, pp. 52–63, 2018.

[19] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," Advances in Neural Information Processing Systems, vol. 30, pp. 3146–3154, 2017.