# TrustConnect: An In-Vehicle Anomaly Detection Framework through Topology-Based Trust Rating

Ayan Roy and Jeetkumar Patel
School of Engineering and Computing
Christopher Newport University
Newport News, Virginia
Email: {ayan.roy, jeetkumar.patel.23}@cnu.edu

Rik Chakraborti
Department of Economics
Christopher Newport University
Newport News, Virginia
Email: rik.chakraborti@cnu.edu

Shudip Datta
Software Engineer
Email: shudipdatta@mst.edu

***Abstract*—Modern vehicles are equipped with numerous in-vehicle components that interact with the external environment through remote communications and services, such as Bluetooth and vehicle-to-infrastructure communication. These components form a network, exchanging information to ensure the proper functioning of the vehicle. However, the presence of false or fabricated information can disrupt the vehicle's performance. Given that these components are interconnected, erroneous data can propagate throughout the network, potentially affecting other components and leading to catastrophic consequences. To address this issue, we propose TrustConnect, a framework designed to assess the trustworthiness of a vehicle's in-vehicle network by evaluating the trust levels of individual components under various network configurations. The proposed framework leverages the interdependency of all the vehicle's components, along with the correlation of their values, and the vulnerability to remote injection based on the outside exposure of each component, to determine the reliability of the in-vehicle network. The effectiveness of the proposed framework has been validated through programming simulations conducted across various scenarios, using a random distribution of an in-vehicle network graph generated with the Networkx package in Python.**

***Index Terms*—Trust Management, In-Vehicle Network, CAN, Anomaly Detection, Security**

## I. Introduction

Modern vehicles today are equipped with numerous electronic control units (ECUs) that manage and execute a wide range of functions critical to vehicle operation [1]. These ECUs oversee essential systems such as autonomous driving, infotainment, safety features, and driver assistance, which significantly contribute to the overall comfort and safety of passengers. The ECUs within a vehicle form an intricate network, often referred to as the *in-vehicle network*, which facilitates communication and information exchange via protocols such as the Controller Area Network (CAN) bus, a defacto standard for in-vehicle communication owing to its stability and cost-effectiveness[2].

Many ECUs are exposed to remote connectivity via Bluetooth, 5G, and other internet services, which increases the risk of remote injection attacks by malicious actors [3][4]. These attacks exploit vulnerabilities in the vehicle's systems [5], with the potential for significant impact, as demonstrated by several researchers [6][7]. A report from 2020 revealed several vulnerabilities that could potentially allow for the remote control of Mercedes-Benz vehicles [8]. A major factor contributing to the success of remote injection attacks is the lack of an authentication mechanism for the information transmitted through the CAN bus [9]. Given the widespread connectivity of modern vehicles, which include numerous sensors, actuators, and ECUs, ensuring the reliability of data from each component is critical. A fault or security breach in any single ECU can propagate across the network, potentially affecting other components and leading to catastrophic failures. As vehicles increasingly depend on complex, interconnected systems, it is essential to assess and preserve the integrity and trustworthiness of the data exchanged within these networks to prevent such vulnerabilities from compromising vehicle safety and functionality.

In this direction, researchers have invested considerable effort in developing frameworks to detect intrusions or anomalies within in-vehicle networks. Common approaches include machine/deep learning models [10], signature-based/sequence systems [11][12], frequency/time-based systems [13], all designed to filter out anomalous data or detect intrusions. These methods can also be utilized to create a framework that assesses the trustworthiness of various ECUs within the network. However, each of these approaches has its limitations. One major drawback of machine learning-based models is their high resource consumption, which is often not feasible for in-vehicle networks. Signature-based systems face the issue that an ECU has a fixed set of outputs, but the timing of those outputs is not considered, making them vulnerable to replay attacks by an attacker during a remote injection. Similarly, frequency and time based systems focus on the timing intervals at which an ECU generates messages, but an attacker can exploit these intervals to deceive the system.

In this paper, we introduce TrustConnect, a framework designed to assess the trustworthiness of an in-vehicle network by evaluating the trust of individual ECUs (Electronic Control Units). The ECUs are represented as a directed graph, enabling the evaluation of each ECU's trust based on its relationships with other ECUs in the network. Furthermore, we account for the potential risks associated with remote exposure, considering the impact of remote injection attacks that could undermine the trust of individual ECUs and compromise the overall integrity of the network. The key contributions of this paper are as follows:

- Proposes a novel trust framework for determining the trustworthiness of an in-vehicle network by calculating the individual trust levels of various ECUs within the network. The proposed model does not rely on predefined rules or signatures to identify anomalous data within the ECUs. Instead, the trust of the network is based on a snapshot of the data generated by different ECUs at a specific point in time.

## II. Related Works

A significant amount of research has been dedicated to filtering out anomalous data by designing Intrusion Detection Systems (IDS) based on various strategies. Jin et al. proposed a signature-based IDS [14], which utilizes factors such as message ID, time, correlation, context, and value range. Although the correlation approach considers the relationships between different components, it does not account for the relative weight or importance of each component. This limitation allows an attacker to exploit a more vulnerable system to deceive a defender into thinking that a less vulnerable system has been compromised. This could occur when the correlation of data from a less vulnerable ECU is used with data from a more vulnerable ECU, misleading the detection system.

A similar correlation-based anomaly detection system was proposed by Xiao et al. [15], which leverages Graph Neural Networks. Their system effectively uses the correlation between changes in traffic bytes and the state of other traffic bytes to identify anomalies. However, like the earlier approach, this system does not consider the varying vulnerability of each component, potentially limiting its effectiveness in certain scenarios.

Tahsin et al. proposed an anomaly detection system [16] that detects anomalous data based on the sequence of messages within an in-vehicle network. While this method is effective in certain contexts, it struggles in the case of multi-point injection attacks. In such cases, attackers can inject messages that mimic data from multiple ECUs simultaneously, preserving the sequence while inserting fabricated data, thus evading detection.

Unlike some commonly used strategies, our proposed framework, TrustConnect, leverages the interdependency of the different ECUs within the in-vehicle network, along with the vulnerability factor of each ECU, to assess the overall robustness of the entire network. TrustConnect enhances the system's resilience against attacks where an attacker targets the most vulnerable component, propagating the attack to make a less vulnerable component appear compromised. Furthermore, the framework places significant emphasis on the outside exposure of each ECU to determine its trustworthiness. This means that even if a strongly connected ECU with minimal exposure is linked to vulnerable ECUs within the network, it is not automatically flagged as anomalous due to injected data in its components.

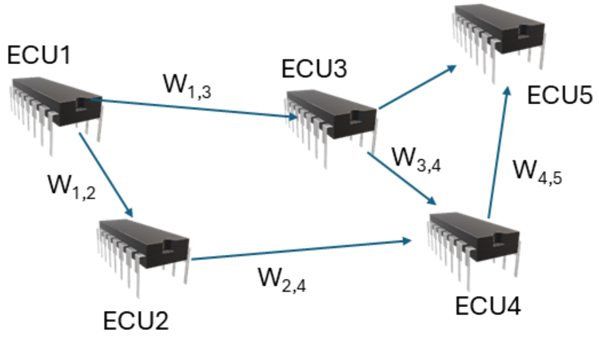


Fig. 1: In-Vehicle ECU Dependency Graph: Each ECU is dependent on other ECUs, with a corresponding weight representing the strength of the dependency.

## III. Preliminaries and Assumptions

In our proposed model, we assume that certain ECUs have greater exposure to the outside environment than others. The more remote the connection to the outside world, the higher the likelihood of a remote injection attack. This likelihood is represented by a value, denoted as $\epsilon_{\rm i}$, for a given ECU $\mathcal{E}_{\rm i}$. A higher value of $\epsilon_{\rm i}$ indicates a stronger connection to the outside world, thereby increasing the vulnerability to a remote injection attack.

## IV. Proposed Framework

In the proposed framework, the in-vehicle network, consisting of various ECUs, is represented as a directed graph as shown in Fig 2. This graph is an ordered pair, denoted as $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of all ECUs within the in-vehicle network, and $\mathcal{E}$ set of edges connecting any two ECUs, $\mathcal{E}_{\rm i}$ and $\mathcal{E}_{\rm j}$. An edge, $\mathcal{E}_{\rm i} \rightarrow \mathcal{E}_{\rm j}$ signifies that the value of $\mathcal{E}_{\rm i}$ can be inferred by $\mathcal{E}_{\rm j}$; in other words, $\mathcal{E}_{\rm i}$ is dependent on the value of $\mathcal{E}_{\rm j}$. Subsequently, we utilize these dependencies among the ECUs within the in-vehicle network to compute the trust score for each ECU, as well as the overall trust score for the entire network. The weight parameter,

### A. Trust Influence of One ECU on Another ECU

For any edge $\mathcal{E}_{\rm i} \rightarrow \mathcal{E}_{\rm j}$, the weight of $\mathcal{E}_{\rm j}$ for $\mathcal{E}_{\rm i}$ is given in eqn 1 and 2 respectively.

$$\mathcal{D}_{\rm i,j} = abs(Val(E_{\rm i,i}) - Val(E_{\rm i,j})) \tag{1}$$

$$\mathcal{W}_{\rm i,j} = e^{-k\cdot\mathcal{D}_{\rm i,j}} \tag{2}$$

As per eqn 1, $\mathcal{D}_{\rm i,j}$ represents the absolute difference between the value observed from $\mathcal{E}_{\rm i}$ and the value that $\mathcal{E}_{\rm j}$ calculates for $\mathcal{E}_{\rm i}$ should have at a given point in time. The primary objective of $\mathcal{D}_{\rm i,j}$ is to quantify the deviation in the reading between a given ECU, $\mathcal{E}_{\rm i}$, its neighboring ECU, $\mathcal{E}_{\rm j}$ on which $\mathcal{E}_{\rm i}$ depends. $k$ is considered a decreasing factor that reduces the weight of $\mathcal{E}_{\rm j}$ on $\mathcal{E}_{\rm i}$ based on the deviation of the values, as calculated in eqn 1.

The weight parameter, $\mathcal{W}_{\mathrm{i,j}}$, represents the variation in the values computed by $\mathcal{E}_{\mathrm{j}}$ for $\mathcal{E}_{\mathrm{i}}$ when the edge $\mathcal{E}_{\mathrm{i}} \rightarrow \mathcal{E}_{\mathrm{j}}$ exists. The role of $\mathcal{W}_{\mathrm{i,j}}$ is to enhance the confidence in $\mathcal{E}_{\mathrm{i}}$ relative to all other $\mathcal{E}_{\mathrm{j}}$s, while also amplifying any deviations that may exist. These deviations may arise when $\mathcal{E}_{\mathrm{j}}$ calculates $\mathcal{E}_{\mathrm{i}}$ based on its inferred values, further influencing the weight parameter's impact.

### *B. Calculating individual ECUs trust and Overall Network trust*

The trust score of each $\mathcal{E}_{\mathrm{i}}$, denoted as $\mathcal{T}_{\mathrm{E(i)}}$, is computed using equation 3. According to this equation, for every $\mathcal{E}_{\mathrm{j}}$ where $\mathcal{E}_{\mathrm{i}} \rightarrow \mathcal{E}_{\mathrm{j}}$ exists, the value of $\epsilon_{\mathrm{j}}$ is incorporated along with a constant, $\alpha$, which gives greater weight to the likelihood of remote injection. Additionally, the term $\mathcal{C}(\mathcal{E}_{\mathrm{j}})$ is included in the equation. When a trust score for $\mathcal{E}_{\mathrm{j}}$ is available, $\mathcal{C}(\mathcal{E}_{\mathrm{j}})$ is replaced by $\mathcal{T}(\mathcal{E}_{\mathrm{j}})$. Consequently, the trust score $\mathcal{T}(\mathcal{E}_{\mathrm{i}})$ for $\mathcal{E}_{\mathrm{i}}$ is calculated using the weight determined in equation 2 for each $\mathcal{E}_{\mathrm{i}} \rightarrow \mathcal{E}_{\mathrm{j}}$ relationship.

$$\mathcal{T}(\mathcal{E}_{\mathrm{i}}) = \sum_{j=1}^{n} (\epsilon_{\mathrm{j}} * \alpha * \mathcal{C}(\mathcal{E}_{\mathrm{j}}) + \mathcal{W}_{\mathrm{i,j}}) \tag{3}$$

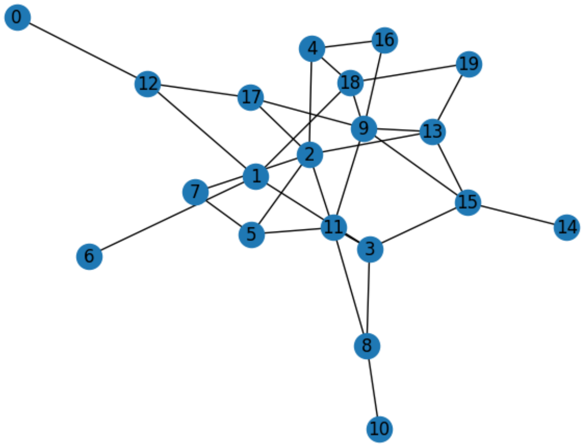


Fig. 2: Simulated In-Vehicle Network Dependency Graph

## V. Experimentation

The performance of the proposed framework has been tested out under different scenarios using python. We utilized the NetworkX package [17] to randomly generate the dependency graph of the in-vehicle network. The evaluation of the proposed framework was simulated using a graph with 20 nodes, where each node represented an ECU, and random connections were established as shown in Figure 1.

During our experiments with various value combinations, we established three key evaluation metrics: *Baseline Trust Value*, *Trust Value*, and *ECU Adjusted Trust Value*. The *Baseline Trust Value* for an $\mathcal{E}_{\mathrm{i}}$ indicates that there is no remote injection in the network and no discrepancy between the value observed from $\mathcal{E}_{\mathrm{i}}$ and the value calculated by $\mathcal{E}_{\mathrm{j}}$ for $\mathcal{E}_{\mathrm{i}}$ in any $\mathcal{E}_{\mathrm{i}} \rightarrow \mathcal{E}_{\mathrm{j}}$ connection. The *Trust Value* for an $\mathcal{E}_{\mathrm{i}}$ is the sum of the values calculated by other $\mathcal{E}_{\mathrm{j}}$s, where an $\mathcal{E}_{\mathrm{i}} \rightarrow \mathcal{E}_{\mathrm{j}}$ connection exists in the in-vehicle network graph, as calculated in equation 3. The *ECU Adjusted Trust Value* for an $\mathcal{E}_{\mathrm{i}}$ represents the adjusted trust score, as shown in equation 4, which is calculated after considering the feasibility of remote injection for an ECU, given by the parameter $\epsilon_{\mathrm{i}}$ (random distribution considered in our evaluation as shown in Fig. 3), and the trust scores calculated in equation 3.

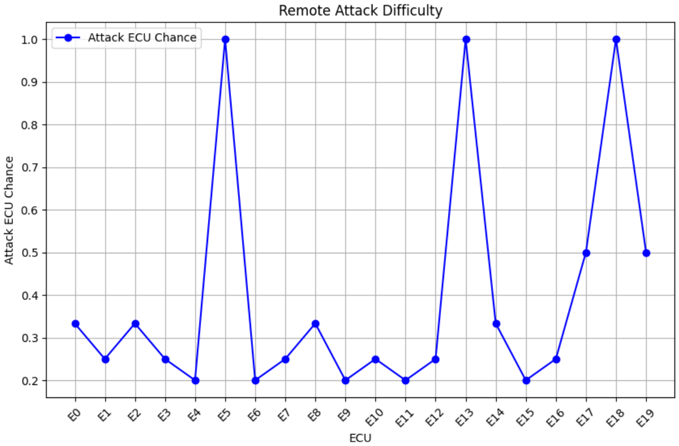


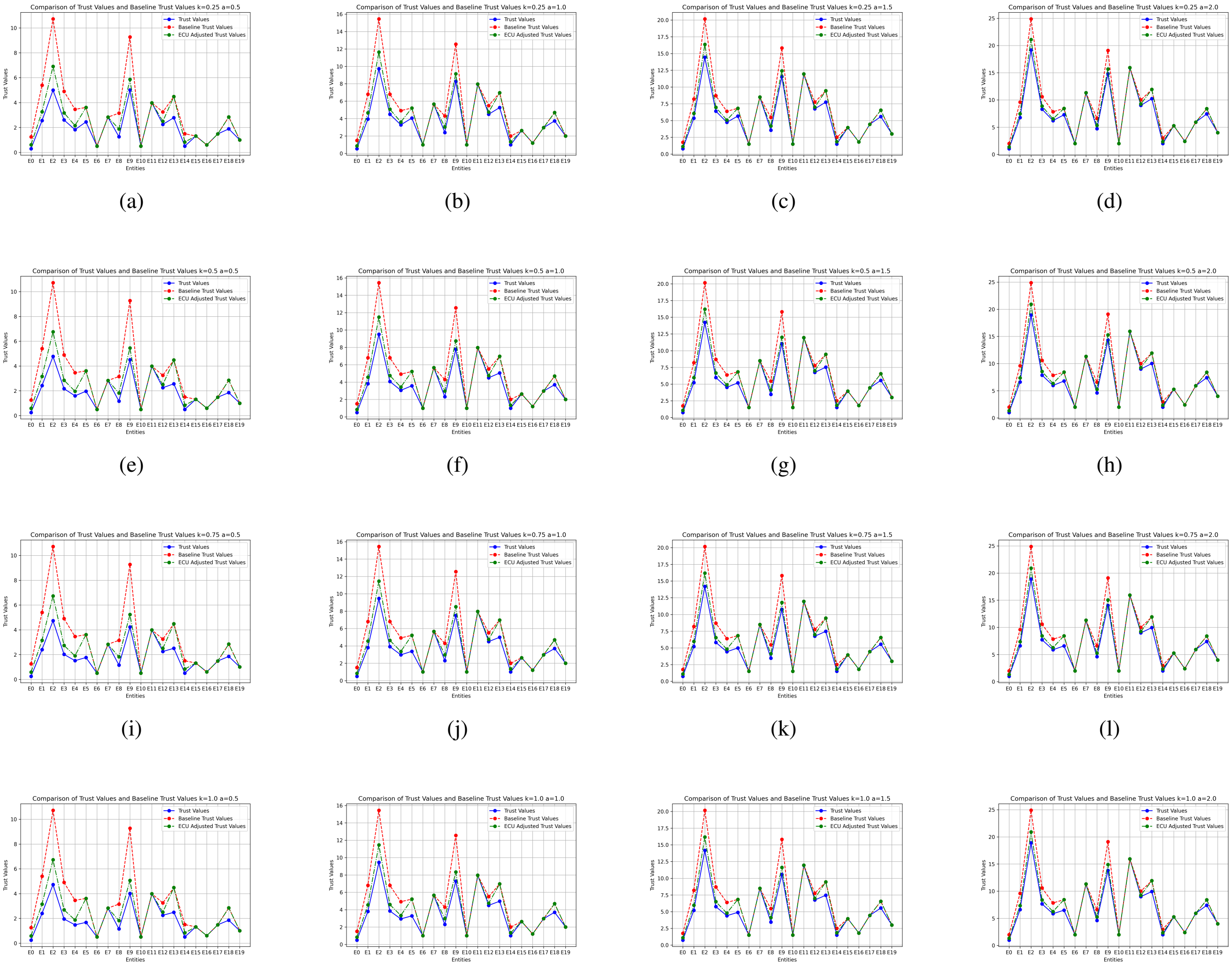
Fig. 3: Simulated Remote Attack Difficulty of Individual ECU-1: Low chance of attack, 0: high chance of attack

$$EATV(\mathcal{E}_{\mathrm{i}}) = BTV(\mathcal{E}_{\mathrm{i}}) - ((BTV(\mathcal{E}_{\mathrm{i}}) - \mathcal{T}(\mathcal{E}_{\mathrm{i}})) * (1 - \epsilon_{\mathrm{i}})) \tag{4}$$

where $EATV, BTV$ is the *Baseline Trust Value and ECU Adjusted Trust Value* of $\mathcal{E}_{\mathrm{i}}$ respectively.

The different combinations of the evaluation metrics talked about in the previous section along along with the impact of $k$ as mentioned in equation 2 and $\alpha$ (denoted as a) from equation 3 has been shown in Fig. 4.

From Fig. 3, it is evident that ECU, E5, E13, and E18 exhibit a low likelihood of being attacked, or in other words, they show greater resilience to remote injection. Additionally, E5, E13, and E18 possess relatively lower connectivity, with 3, 4, and 4 edges, respectively, as shown in the in-vehicle network in Figure 2. Due to this relatively low connectivity, the *Baseline Trust Values* for these nodes, across all combinations of $k$ and $\alpha$ (represented as 'a' in Figure 4), remain relatively low. It is also observable across all the various combinations in Figure 4 that the *ECU Adjusted Trust Value* for E5, E13, and E18 is almost identical to their *Baseline Trust Values*, despite the *Trust Values* based on the other connected ECUs being lower. This outcome occurs because the ECUs connected to each of them exhibit a significantly higher likelihood of a remote injection attack, reducing the reliability of information from ECUs that are more vulnerable to injection attacks, compared to an ECU with a lower attack likelihood. As a result, by using eqn 4, the *ECU Adjusted Trust Value* remains relatively



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

Fig. 4: Comparison of the *Baseline Trust Value*, *Trust Value*, and *ECU Adjusted Trust Value* under different combinations of the k and $alpha$ (denoted as a) values

close to the *Baseline Trust Values*. In contrast, the disparity between the *Baseline Trust Values* and the *ECU Adjusted Trust Value* is noticeably larger for E2 and E9 (as shown in Figure 4). This difference is attributed to the relatively higher chance of remote injection attacks on E2 and E9 (as indicated by Figure 3). E2 is connected to E5, E17, E1, E11, E4, and E13, of which E5, E13, and E17 have a much lower chance of remote injection. Therefore, in the case of any discrepancies in the data calculations, the proposed framework can flag E2 as a potential source of attack, warranting further investigation to identify any additional vulnerabilities in the system. Thus, from the proposed framework it is to be noted that even the exposure of the different ECUs to the outside world play a significant role in determining whether there is a chance of a remote injection or not. If multiple vulnerable components are connected to a resilient component, as seen above in the example of E18, E13, and E5, it may not warrant an immediatiate investigation. This enhances the framework's resilience in scenarios where an attacker might intentionally try to deceive the system by fabricating a false scenario in which a resilient component appears to be under attack, while the actually compromised vulnerable components connected to the resilient one are overlooked. However, it is important to note that the system can lower the score of a resilient ECU if a sufficient number of ECUs contradict its readings, or if highly resilient ECUs contradict one another. In such cases, an Intrusion Detection System (IDS) can be designed to trigger an investigation. Such an IDS could be designed to trigger an investigation when the threshold or frequency of contradictory ECUs exceeds a certain level. The IDS would consider both the number of ECUs that contradict each other and the resilience of the contradicting ECUs to remote injection attacks, thereby enabling a more comprehensive evaluation of the data integrity from the entire in-vehicle network.

It can also be observed from the results in Figure 4 that the effect of the decay rate, $k$, in the trust calculation as defined in equation 2, has a minimal impact on the trust values. This is because the value of $k$ is relatively small in the experiment, so any changes to it result in only a minor effect on the overall trust value. A potential solution to this limitation could be to consider a higher value for $k$ or to apply a logarithmic function to equation 1 to amplify the distance, though such approaches fall outside the scope of this study. Additionally, the value of $\alpha$ (denoted as 'a' through subfigures a-p in Figure 4) significantly influences the *Trust Value* for each ECU. This is due to the amplification of the remote injection probability, $\epsilon$, for each connected ECU in equation 3, which in turn contributes to changes in the 3 values of all dependent ECUs.

## VI. Conclusion

In this paper, we have proposed a trust framework that assesses the trustworthiness of individual ECUs within an in-vehicle network, based on their connectivity to the outside world. By leveraging the interconnectivity of the ECUs within the in-vehicle network, the framework determines the vulnerability of an ECU to attack, considering the connectivity of the ECUs it is linked to. This framework is effective for analyzing and studying the scope and propagation of remote injection attacks, taking into account the ECUs' connectivity within the network. Additionally, the proposed framework can be utilized to design an efficient Intrusion Detection System (IDS) that does not rely on statistical data, the frequency of information received from the ECUs, or any data signatures that could be exploited by an attacker to replay or manipulate communication within the network. The framework can also help identify reliability issues within the in-vehicle system, even if an attacker exploits data patterns from specific ECUs or manipulates the frequency of injected data.

## References

[1] Jean Walrand, Max Turner, and Roy Myers. An architecture for in-vehicle networks. *IEEE Transactions on Vehicular Technology*, 70(7):6335–6342, 2021.

[2] Hyun Min Song and Huy Kang Kim. Self-supervised anomaly detection for in-vehicle network using noised pseudo normal data. *IEEE Transactions on Vehicular Technology*, 70(2):1098–1108, 2021.

[3] Shu Nakamura, Koh Takeuchi, Hisashi Kashima, Takeshi Kishikawa, Takashi Ushio, Tomoyuki Haga, and Takamitsu Sasaki. In-vehicle network attack detection across vehicle models: A supervised-unsupervised hybrid approach. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 1286–1291. IEEE, 2021.

[4] Ayan Roy and Sanjay Kumar Madria. Cansafe: An mtd based approach for providing resiliency against dos attack within in-vehicle networks. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 3243–3250, 2022.

[5] Rajkumar Singh Rathore, Chaminda Hewage, Omprakash Kaiwartya, and Jaime Lloret. In-vehicle communication cyber security: challenges and solutions. *Sensors*, 22(17):6679, 2022.

[6] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno. Comprehensive experimental analyses of automotive attack surfaces. In *20th USENIX security symposium (USENIX Security 11)*, 2011.

[7] Charlie Miller and Chris Valasek. Adventures in automotive networks and control units. *Def Con*, 21(260-264):15–31, 2013.

[8] Minrui Yan, Jiahao Li, and Guy Harpak. Security research report on mercedes-benz cars. In *Black Hat USA 2020*, page 38, 2020. Accessed: 2024-11-30.

[9] Heng Sun, Miaomiao Chen, Jian Weng, Zhiquan Liu, and Guanggang Geng. Anomaly detection for in-vehicle network using cnn-lstm with attention mechanism. *IEEE Transactions on Vehicular Technology*, 70(10):10880–10893, 2021.

[10] Xu Wang, Yihu Xu, Yinan Xu, Ziyi Wang, and Yujing Wu. Intrusion detection system for in-vehicle can-fd bus id based on gan model. *IEEE Access*, 2024.

[11] Mirco Marchetti and Dario Stabili. Anomaly detection of can bus messages through analysis of id sequences. In *2017 IEEE intelligent vehicles symposium (IV)*, pages 1577–1583. IEEE, 2017.

[12] Chao Wang, Xueqiao Xu, Ke Xiao, Yunhua He, and GuangCan Yang. Traffic anomaly detection algorithm for can bus using similarity analysis. *High-Confidence Computing*, page 100207, 2024.

[13] Adrian Taylor, Nathalie Japkowicz, and Sylvain Leblanc. Frequency-based anomaly detection for the automotive can bus. In *2015 World Congress on Industrial Control Systems Security (WCICSS)*, pages 45–49, 2015.

[14] Shiyi Jin, Jin-Gyun Chung, and Yinan Xu. Signature-based intrusion detection system (ids) for in-vehicle can bus network. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2021.

[15] Junchao Xiao, Lin Yang, Fuli Zhong, Hongbo Chen, and Xiangxue Li. Robust anomaly-based intrusion detection system for in-vehicle network by graph neural network framework. *Applied Intelligence*, 53(3):3183–3206, 2023.

[16] Tahsin CM Dönmez. Anomaly detection in vehicular can bus using message identifier sequences. *IEEE Access*, 9:136243–136252, 2021.

[17] Aric Hagberg and Drew Conway. Networkx: Network analysis with python. *URL: https://networkx. github. io*, 2020.