Auto-CIDS: An Autonomous Intrusion Detection System for Vehicular Networks

Mohsen Sorkhpour
Cyber Science Lab (CSL), Canada
Cyber Foundry (CCF), University of Guelph
Guelph, ON, Canada
sorkhpurmohsen@gmail.com

Abbas Yazdinejad
Cyber Science Lab (CSL), Canada
Cyber Foundry (CCF), University of Guelph
Guelph, ON, Canada
ayazdine@uoguelph.ca

Ali Dehghantanha
Cyber Science Lab (CSL), Canada
Cyber Foundry (CCF), University of Guelph
Guelph, ON, Canada
adehghan@uoguelph.ca

Abstract

Control Area Network (CAN), despite facilitating electronic control unit (ECU) communications, lacks built-in mechanisms for secure transmission, exposing its messages to cyber-attacks due to unsecured broadcasting. Current Intrusion Detection Systems (IDSs) for CAN rely predominantly on rule-based, statistical, or supervised machine learning (ML) models, which require significant human intervention for tasks such as reconfiguration, gathering labeled data samples, and retraining with newly released vehicle models. These manual dependencies highlight the critical need for autonomous capability in IDS that can adapt independently, thus mitigating practical deployment challenges in real-world scenarios. In this paper, we propose an autonomous cybersecurity IDS named Auto-CIDS, designed to minimize human intervention and enable active learning utilizing past experiences. By applying Deep Reinforcement Learning (DRL) with the advantages of unsupervised algorithms, we train Deep Q-network (DQN) agents in a self-supervised manner using their own past experiences. We develop three standalone autonomous methods. The first method, Single-Task Self-Supervised, uses an autoencoder to supervise DQN agents in each environment, which includes both normal and specific attack data without needing labeled datasets. The second method, Multi-Environment Self-Supervised, enhances the generalization ability of the first by training a DQN agent across multiple environments, allowing knowledge transfer from varied settings into a single agent. The third method, Multi-Task Multi-Agent, increases the robustness of our proposed Auto-CIDS by employing a combination of modified unsupervised methods, including autoencoder, k-means, and isolation forest algorithms, each tailored for a specific type of attack. This approach builds attack-specific DQNs that periodically and cooperatively train a global DQN agent based on their predictions, facilitating ongoing active learning. We conducted experiments on the Car-Hacking dataset, which includes Denial of Service (DoS), fuzzy, and spoofing attacks, and the results demonstrate the effectiveness of these methods in detecting cyber-attacks. The results also demonstrate high evaluation metrics, including False Negative Rate (FNR), Error Rate (ER), accuracy, recall, precision, and F1 score. Additionally, these methods significantly reduce the need for human intervention by enhancing the autonomy of our proposed IDS. This increased autonomy enables the systems to adapt to new environments, thereby making our Auto-CIDS more autonomous and adaptable.

CCS Concepts

• **Security and privacy → Systems security and Security services**.

Keywords

Deep Reinforcement Learning, RL, DQN, Autonomous Cybersecurity, IDS, CAN.

ACM Reference Format:
Mohsen Sorkhpour, Abbas Yazdinejad, and Ali Dehghantanha. 2024. Auto-CIDS: An Autonomous Intrusion Detection System for Vehicular Networks. In *Proceedings of AutonomousCyber 2024 (1st International Workshop on Autonomous Cybersecurity)* , 11 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

The automotive industry has transitioned from mechanical systems to advanced, interconnected computing platforms, driven by the integration of embedded ECUs that manage critical functions such as engine control and advanced driver-assistance systems (ADAS) [15]. This shift has introduced new cybersecurity vulnerabilities, particularly within the CAN protocol, which lacks built-in security mechanisms. Moreover, increased connectivity through technologies like Bluetooth, WiFi, and telematics has further exposed vehicles to cyber-attacks [3].

Recent works [16], [10], [28], [11] have revealed vulnerabilities in automotive communication systems, especially the CAN bus. These studies show how attackers can exploit both physical and remote access points to compromise vehicle safety. The inherent lack of authentication and encryption in the CAN bus significantly heightens its vulnerability. Attackers can use these vulnerabilities to send malicious commands to ECUs, causing dangerous situations like sudden braking or engine shutdowns. These issues highlight the need for a comprehensive approach to secure the CAN bus against cyber attacks, making it imperative to develop a robust IDS framework to identify and prevent such attacks. An in-depth examination of prior research on CAN IDS is provided in [3]. These studies consistently underline the necessity of human intervention

1st International Workshop on Autonomous Cybersecurity, Oct 2024, Salt Lake City, U.S.A

ACM ISBN XXXXX-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

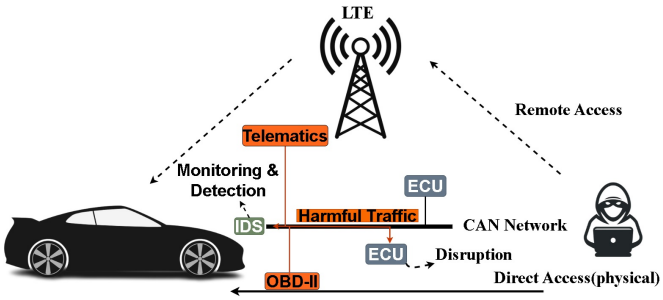

Figure 1: Typical message injection attack scenarios

in these IDSs. Also, current systems predominantly utilize rule-based, statistical, and supervised ML techniques [25], [21], which are not well-suited to the dynamic nature of the CAN protocol. The variability in specifications, such as CAN ID message frequency patterns across different vehicle manufacturers, complicates the application of existing IDSs to newly released models. These systems typically require expert knowledge for ongoing maintenance, with manual tasks including extensive data labeling, retraining, and reconfiguration to accommodate new CAN specifications. This need significantly increases maintenance overhead and limits the practical deployment of current IDSs in real-world scenarios. Thus, there is a critical need for an innovative approach that can minimize human intervention and autonomously adapt to new environments, addressing these longstanding challenges.

To address these challenges, we propose Auto-CIDS, an autonomous IDS leveraging unsupervised learning and DRL. Our system minimizes human intervention by enabling DQN agents to learn from past experiences and adapt to new environments through self-supervised mechanisms. We introduce three novel methods: Single-Task Self-Supervised, Multi-Environment Self-Supervised, and Multi-Task Multi-Agent approach, each enhancing the system's autonomy and detection capabilities. This system enhances detection capabilities while minimizing the need for human intervention, perfectly aligning with the dynamic nature of vehicular communication systems. Specifically, we implement a DRL algorithm [18] that allows DRL agents to learn from past experiences, enabling them to adapt to new CAN specifications with minimal human input. Furthermore, to enhance the autonomy of our methods, we incorporate self-supervised DRL approaches that autonomously mitigate the CAN limitations. By leveraging self-supervised learning, our IDS effectively utilizes the unlabeled data from the vehicle's CAN bus to detect potential cyber attacks, thereby reducing reliance on extensively labeled datasets. These advancements not only decrease the dependency on labeled data but also enhance the system's ability to adjust to new vehicle models and emerging threats, increasing the scalability and practicality of our IDS for real-world applications. The primary contributions of this work can be summarized as follows:

- For the first time, we design and implement **Auto-CIDS**, an autonomous cybersecurity IDS specifically designed for vehicular networks, which dynamically adapts to new conditions during runtime. Utilizing DRL, particularly the DQN, Auto-CIDS minimizes the need for human interaction. The proposed system integrates unsupervised ML algorithms as supervisors, enabling it to autonomously process and learn from data that these algorithms automatically label. This capability allows our system to continuously and actively learn, significantly enhancing the adaptability and scalability of the IDS in real-world vehicular network environments.
- We introduce the **Multi-Environment Self-Supervised method** to enhance the generalization of the policies obtained across diverse environments in vehicular networks. This approach enables a single DRL agent to handle a broad spectrum of attacks by exposing it to various environments encompassing all the attacks discussed in this paper. Such exposure not only improves generalization and system adaptability but also reduces human intervention, thereby facilitating easier deployment with a single agent.
- Our research introduces the **Multi-Task Multi-Agent Self-Supervised Approach**, achieving full autonomy through diverse unsupervised ML algorithms. Leveraging the collective knowledge of multiple attack-specific, self-supervised DRL agents allows for continual adaptation and improvement without direct human intervention. This ongoing enhancement is crucial for maintaining effectiveness against evolving cyber threats and diversity in the CAN protocol, making it ideally suited for long-term deployment in real-world scenarios.
- To evaluate our proposed method, we utilized actual vehicle data from the **Car-Hacking dataset**. Additionally, we tested the robustness and adaptability of our methods under unseen conditions using a synthetic dataset designed to simulate unknown environments. We demonstrate the robustness of our system against diverse attacks and its autonomous adaptation to new and previously unseen environments, underscoring its potential for long-term deployment across various CAN protocols.

These contributions collectively underscore the necessity and effectiveness of autonomous cybersecurity IDS in enhancing the safety and security of modern vehicular networks. Our approach demonstrates significant advancements in the autonomous detection and mitigation of cyber-attacks, paving the way for more secure vehicle-road cooperation systems.

The remainder of this study is organized as follows: Section 2 provides background and related work about in-vehicle network security and unsupervised methods. Section 3 introduces the proposed Auto-CIDS and methodological steps. Section 4 presents the evaluation metrics and experiment results. Finally, we conclude this study in Section 5.

2 Related Work

As modern vehicles advance towards autonomous driving and sophisticated network communications, securing in-vehicle networks grows increasingly crucial. Key strategies against in-vehicle cyber-attacks are message authentication and intrusion detection [5], [14],[4],[17]. Message authentication provides higher security levels despite requiring more resources and time. Conversely, current

IDSs in CAN primarily rely on rule-based and supervised ML methods. Supervised ML methods heavily rely on human intervention and expert-labeled data, which makes the real-world deployment of current IDSs more challenging [29].

Authors in [22] proposed a frequency-based, lightweight IDS for the CAN bus to detect vehicle data injection attacks. The system relies on the time intervals between CAN data frames for anomaly detection. Similarly, [24] proposed an entropy-based anomaly detection method, defining entropy on the CAN bus and detecting attacks by comparing the entropy to a reference. In [12], authors propose an IDS that enhances attack detection efficiency by exploiting malicious patterns using a threshold obtained through a brute-force optimization algorithm. Researchers in [26] discovered that the time interval between messages is a significant feature for detecting attacks in CAN traffic. They introduced a lightweight IDS for in-vehicle networks, utilizing a time interval analysis (TIA) of CAN messages. All these methods suffer from changing the specification of CAN protocols.

On the supervised ML side, authors in [1] proposed an effective IDS scheme for binary classification that utilizes eight supervised ML algorithms along with ensemble classifiers. They found that ensemble classifiers achieved better accuracy than individual models, as ensemble learning strategies have superior performance through a combination of multiple learning mechanisms. Authors in [27] proposed a malicious traffic classification method using a convolutional neural network (CNN). As the network traffic data was processed as an image and inputted to their CNN, raw network traffic was used, eliminating the need for hand-designed features. Similarly, in [23], authors proposed another CNN-based IDS that utilizes a data frame of 29 sequential CAN IDs to detect patterns on the CAN bus. For supervised learning, frames containing injected messages are labeled as attacks, while those without are labeled as non-attacks. Authors in [9] proposed an anomaly detection model for CAN bus traffic called MOCSVM. They introduced a modified version of the one-class support vector machine (OCSVM), which helps prevent the algorithm from becoming trapped in local optima and avoids premature convergence. In [2], authors proposed a deep learning-based IDS to detect malicious attacks based on the human behavior of attackers. The algorithm applies multilayer perceptrons (MLP) to train the algorithm. Nevertheless, the diverse specifications of CAN protocols across vehicle manufacturers demand frequent reconfiguration or retraining of these systems, which leads to increased maintenance overhead of current IDSs. Moreover, in [30] has proposed a kangaroo-based intrusion detection system (KIDS), leveraging software-defined networking flexibility and programmability for scalable and efficient attack detection. The KIDS architecture enhances performance by using a zone-based design and monitoring packet parsers and flow tables, demonstrating improved detection of malicious packets.

Despite the advancements in IDS methodologies, several critical limitations persist. Existing IDS methods, whether rule-based, statistical, or supervised ML models, suffer from high dependency on labeled data, frequent reconfiguration, and extensive manual maintenance. These approaches also struggle with generalization and adaptability across different vehicle models and attack scenarios. Furthermore, the lack of active learning capabilities and collaborative learning mechanisms hinders their ability to effectively reuse past experiences and adapt to new threats autonomously. Addressing these research gaps emphasizes the necessity of developing an autonomous IDS that can significantly reduce human intervention, enhance generalization and adaptability, facilitate active learning, and promote collaborative learning. Such an autonomous system is essential to ensure the integrity and safety of vehicular networks in dynamic and evolving environments.

3 Problem Statement

The primary challenge for a supervised IDS is gathering sufficient labeled data samples, especially for newly released car models. Almost all previous IDS implementations heavily rely on CAN IDs to learn the normal frequency of each CAN ID [22]. Let $\text{IDS}_{\text{supervised}}$ represent a supervised IDS model trained on dataset D, containing labeled CAN messages M_i associated with CAN IDs ID_i. The model learns the normal frequency $\text{Freq}_{\text{normal}}(\text{ID}_i)$ and distinguishes between normal ($\text{N}(ID_i)$) and abnormal ($\text{A}(ID_i)$) frequency patterns. However, transferring $\text{IDS}_{\text{supervised}}$ from dataset D_A to D_B proves challenging due to unique frequency patterns in CAN IDs ($\text{Freq}_{\text{normal}}(\text{ID}_i)_{D_A} \neq \text{Freq}_{\text{normal}}(\text{ID}_i)_{D_B}$) across vehicle models [8]. This lack of generalization necessitates manual retraining for each vehicle model, leading to scalability and efficiency issues. Consequently, the majority of supervised IDSs lack generalization, requiring extensive data collection and retraining for each vehicle model.

To address this, we need to have an autonomous adaptation mechanism, where $\theta' = \arg\min_\theta \mathcal{L}(\text{IDS}_{\text{unsupervised}}(\theta, D'), D')$, allowing the unsupervised IDS to adapt to new datasets D' without manual intervention autonomously. This approach leverages the data-driven insights from the vehicle's CAN bus to dynamically update the IDS parameters θ, minimizing the discrepancy in model performance across different vehicle models and enhancing the system's ability to respond to new and evolving threats. Therefore, autonomous methods are crucial to adapt $\text{IDS}_{\text{supervised}}$ to new models, overcoming the limitations imposed by the specificity of CAN IDs.

3.1 Threat Model

To clearly delineate and understand the various aspects of attacks within in-vehicle networks, we systematically formulate each type of attack similar to [7, 13, 19]. Figure 1 illustrates the common message injection attacks encountered in these networks, providing a visual representation that aids in comprehending the potential threat scenarios.

Input Representation: Consider a vehicular system that communicates through a CAN bus. The CAN messages at time t can be represented as follows:

$$M(t) = (T(t), ID(t), DLC(t), DF(t))$$

These components include the timestamp $T(t)$, the CAN identifier $ID(t)$, the data length code $DLC(t)$, and the data frame $DF(t)$ containing the payload. However, we have only used the DF as the input; therefore, Under normal conditions, the data frame $DF_{\text{normal}}(t)$ is a function of the vehicle's state variables and control inputs:

$$DF_{\text{normal}}(t) = f(V(t), C(t))$$

ECUs produce data related to vehicle state and control inputs, so in the formula above, the $V(t)$ represents vehicle state variables such as speed and engine RPM, and the $C(t)$ represents control inputs such as throttle position and brake pressure.

Attack Effects: In a CAN-bus system, various attacks can target specific functionalities, such as an attacker changing the gear by injecting messages with a specific CAN identifier related to the gear function. Each attack contributes to the deviation from normal behavior when multiple attacks occur. The show effect of all attacks on the data frame $DF(t)$ is represented as $DF_{total}(t)$:

$$DF_{total}(t) = \sum_{i}^{N} \lambda_i \cdot U_i(t) + DF_{\text{normal}}(t)$$

In this formula, N represents the total number of attack types. Each $\lambda_i(t)$ represents the deviation caused by a specific type of attack i at time t, and $U_i(t)$ is the step function that indicates the duration of the attack type i at time t. Therefore, deviations can vary depending on the type of attack. This formulation $DF_total(t)$ captures the total CAN-bus traffic, including the impact of various attacks, by aggregating the individual deviations caused by each type of attack plus the baseline traffic DF_{normal} that exists on the CAN-bus even in the absence of any attacks. The overall deviation in the data frame $\Delta DF(t)$ due to all attacks can be formulated as:

$$\Delta DF(t) = DF_{total}(t) - DF_{\text{normal}}(t)$$

The objective is to detect these deviations to enhance the security and reliability of vehicular communication systems against cyber-attacks. This involves continuously tracking CAN messages and identifying significant deviations from expected patterns. We aim to dynamically define a boundary ∂ through interactions with the environment using a DRL agent. Let $D \subset \mathbb{R}^n$ be a region with boundary ∂D. Define $A \subset D$ as the set of points where $\Delta DF(t)$ is positive and not within the learned boundary ∂D:

$$A = \{\Delta DF(t) \mid \Delta DF(t) > 0,\ \Delta DF(t) \notin \partial D\}$$

Here, ∂D is adaptively learned by the DRL agent based on observed data and system state interactions. Detecting these deviations allows for proactive measures to mitigate potential cyber-attacks on vehicular communication systems.

3.2 Design Goal

This work aims to innovate autonomous features in IDS by establishing unique design goals that reduce human intervention, enhance generalization and adaptability, facilitate active learning, and leverage collaborative learning through continuous CAN-bus traffic monitoring.

- **Reducing human intervention:** Essential in dynamic environments like CAN, our design reduces the need for human intervention by automating retraining and reconfiguration processes. This allows the IDS to adapt to protocol changes across different vehicle models swiftly, eliminating the need for manual adjustments.
- **Generalization and adaptability:** Our IDS strives for robustness by generalizing across various attacks. By transferring knowledge from diverse environments to a single decision-making agent, the system effectively learns and applies policies from one scenario to another, facilitating detection and mitigation in new and unfamiliar environments. This is particularly crucial given the unique CAN ID message frequency patterns that vary by vehicle model and manufacturer.
- **Active learning and reusing past experiences:** The proposed IDS integrates active learning strategies that allow DRL agents to continually refine their detection capabilities by leveraging past experiences stored in memory buffers. These experiences, captured during real-time CAN-bus traffic monitoring, provide critical insights into normal and abnormal behaviors, enabling the system to fine-tune detection thresholds based on historical data.
- **Collaborative learning:** Given the complexity and diversity of attacks, relying on a single unsupervised algorithm for supervision is insufficient. We propose a collaborative learning approach where multiple specialized agents, each focusing on different types of attacks such as anomaly detection, cluster analysis, or outlier detection, contribute to a unified training process. These agents share their findings with a central agent, which synthesizes and learns from this collective knowledge. This collaborative effort enhances the central agent's ability to understand and respond to diverse attack patterns, thereby improving detection efficacy.

4 Proposed System

The underlying idea behind the proposed Auto-CIDS, as presented in Figure 2, is to deploy a DRL agent that autonomously detects attack messages in the CAN-bus. To enhance the autonomy of our Auto-IDS, we utilize unsupervised ML algorithms to supervise the DRL agents. Our methods are designed to operate without human intervention, providing robust and adaptive defense mechanisms against cyber attacks targeting the CAN-bus system. We introduce three meticulously crafted methods to intensify the autonomy levels of CAN-bus intrusion detection. This shift towards autonomous learning signifies a crucial step forward in ensuring the security of vehicular networks in the face of unseen conditions. In the following subsections, we cover the foundational aspects relevant to our research before diving into the specifics of each approach, including their architectures and the intricacies of their training phases.

4.1 Adapting DRL for CAN-Bus IDS

This section is dedicated to outlining the common aspects of the proposed methods. All DQN agents in the proposed methods follow a similar fully connected (FC) architecture but vary in the training phase. The FC architecture comprises 64 inputs and two outputs, with two hidden layers, each comprising 512 and 128 neurons, respectively. The DQN agent is trained by interaction with the environment. Upon correctly identifying an attack, it receives a positive reward (+1); conversely, incorrect identifications yield negative rewards (-1). Based on these rewards, the agent's policies (neuron weights) are updated iteratively until convergence is achieved. The RL components in this research are described in detail below:

- **State:** Each record from the Car-Hacking dataset contains data related to an ECU, including a *Time Stamp, CANID, DLC, Data Frame*, and *label*. We have used the Data Frame

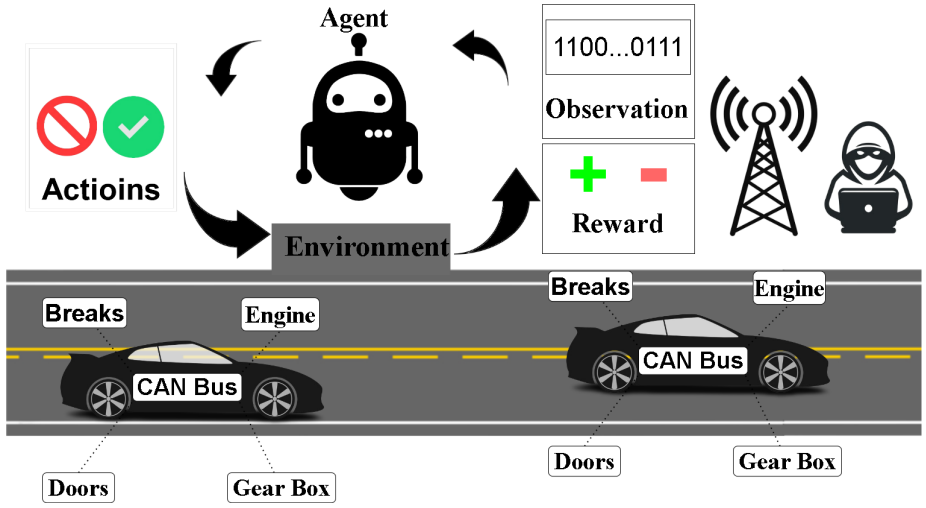

Figure 2: General Overview of Auto-IDS

section as the input of our proposed methods. The Data Frame consists of 8 hexadecimal sections, which we convert into binary data to enhance the agent's discernibility and sensitivity. Thus, each data sample comprises a 64-bit binary array, representing a state across all proposed methods.

- **Action:** As mentioned, the architecture of all agents remains constant, featuring two outputs for blocking or confirming transmitted data. Therefore, two actions are considered for the agent.
- **Reward:** The reward function is crucial to any reinforcement learning (RL) method. The functionality of this function varies based on the proposed methods, as elaborated in the descriptions of each method. In the subsequent sections, we will explore each proposed method in detail, including their autonomous training processes, to comprehensively understand their effectiveness in CAN-Bus attack detection.

4.2 Proposed Methods

This part of the study focuses on how we have introduced our proposed methods in the Auto-CIDS framework. These methods range from single-task to multi-environment and multi-task approaches, all trained using a self-supervised learning approach. They progressively intensify the autonomy levels of intrusion detection to Reinforce the resilience against attacks targeting the CAN-Bus and effectively adapt to changing CAN specifications, thereby reducing the need for human intervention.

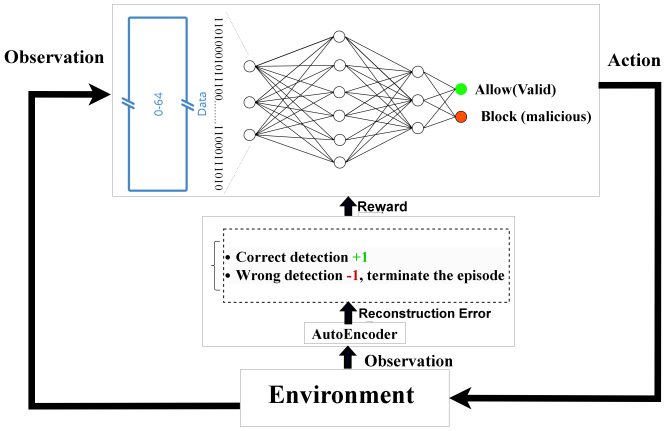

Figure 3: Single-Task Self-Supervised Approach Diagram

4.2.1 ***Single-Task Self-Supervised Approach.*** Our first proposed method, depicted in figure 3, employs a pre-trained autoencoder to supervise the DQN agent. Let S denote the state space representing the observations from the CAN-bus system, where each state s is a binary array representing the data frame section of the CAN messages. Let $A = \{a_1, a_2\}$ represent the action space, where a_1 corresponds to approving the transmission (normal) and a_2 corresponds to blocking it (abnormal). The environment dynamics are represented by the transition function $T : S \times A \rightarrow S$, which describes how the state changes in response to the agent's actions. The reward function $R : S \times A \rightarrow \mathbb{R}$ is defined as follows:

$$R_{\text{self-supervised}}(s, a) = \begin{cases} r_{\text{positive}} & \text{if } a = a_1 \text{ and } \text{RE}(s) < \text{T} \\ r_{\text{negative}} & \text{if } a = a_2 \text{ and } \text{RE}(s) \geq \text{T} \end{cases}$$

where r_{positive} is the positive reward assigned for correctly identifying normal data, r_{negative} is the negative reward for incorrectly classifying abnormal data, and $\text{RE}(s)$ denotes the reconstruction error of the state s obtained from the autoencoder, the threshold for

abnormality detection is calculated as follows: **T = average_error + std_dev** where **average_error** is the average reconstruction error obtained from passing normal dataset instances through the autoencoder, and **std_dev** is the standard deviation of the reconstruction errors. The objective is to learn a policy $\pi : S \rightarrow A$ that maximizes the expected cumulative reward:

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R_{\text{self-supervised}}(s_t, a_t) \right]$$

where γ is the discount factor, s_t is the state at time step t, and $a_t = \pi(s_t)$ is the action selected by the policy. The "Single-Task Self-Supervised Approach" in the formulation consists of the following steps:

- **Training Autoencoder:** In this step, an autoencoder will be trained using the normal dataset to learn the underlying patterns of benign CAN-Bus data.
- **Threshold Calculation:** Next, to build the reward function used in the RL training phase, all normal dataset instances are passed through the trained autoencoder to calculate the average reconstruction error and its standard deviation. Using the formula mentioned above, set the threshold for abnormality detection.
- **DRL with DQN Agent:** In this step, a DQN agent is trained by interacting with the simulated environment. At each time step, the agent observes the current state s, passes it through the autoencoder to obtain the reconstruction error, and selects an action a based on the trained policy derived from the reward function. Finally, the DQN algorithm learns the policy π that maximizes the expected cumulative reward.

By incorporating the autoencoder-based self-supervised mechanism into the DRL, the agent learns to autonomously detect and respond to abnormal data instances, thereby enhancing the security of the CAN-Bus communication.

4.2.2 ***Self-Supervised Multi-Environment Approach.*** We utilized the Self-Supervised Multi-Environment Approach to effectively generalize policies obtained during training across various environments with diverse attack scenarios. This method comprises two crucial phases: a self-supervised pre-training phase and a supervised fine-tuning phase. Implementing this two-phase approach amplifies the system's adaptability and generalization to new environments while minimizing the need for human intervention. Moreover, it significantly accelerates convergence compared to the single-task self-supervised learning method, albeit still requiring a minimal labeled dataset.

Pre-Training with Autoencoder. In the pre-training phase, the DQN agent interacts with an environment where rewards are calculated using unlabeled data. This allows the DQN agent to learn initial representations and patterns through self-supervised learning. These pre-trained representations are a robust foundation for fine-tuning with limited labeled data, thereby improving the model's generalization and adaptation abilities. Consequently, the DQN agent becomes more proficient in identifying and addressing various attack types within automotive cybersecurity contexts. The formula below shows the pre-train phase, where $R_{\text{self-supervised}}(s, a)$ is the reward function specific to multi-environment training.

$$J_{\text{pre-train}}(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R_{\text{self-supervised}}(s_t, a_t) \right]$$

Fine-Tuning with Limited Labeled Data. During the pre-training phase, the model leverages unlabeled data to learn initial representations and patterns via self-supervised learning.

$$R_{\text{fine-tune}}(s, a) = \begin{cases} r_{\text{positive}} & \text{if } a = a_1 \text{ and label}(s) = \text{a}_1 \\ r_{\text{negative}} & \text{if } a \neq a_2 \text{ and label}(s) = \text{a}_2 \end{cases}$$

$$J_{\text{fine-tune}}(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R_{\text{fine-tune}}(s_t, a_t) \right] + J_{\text{pre-train}}(\pi)$$

where $R_{\text{fine-tune}}(s, a)$ is the reward function specific to fine-tuning with limited labeled data, focusing on improving detection accuracy with the available labeled instances. By combining self-supervised learning with multi-environment training and fine-tuning on limited labeled data, the DQN agent can effectively detect and mitigate multiple attack types in real-world automotive cybersecurity scenarios. Figure 4 depicts the self-supervised multi-environment method, providing detailed visual information about its implementation.

4.2.3 ***Multi-task Multi-Agent Self-Supervised Approach.*** Due to the limitations of using autoencoders for tracking all types of attacks, as identified in our investigations of the previous methods, we propose a more versatile approach: the Multi-task Multi-Agent Self-Supervised Approach. This method addresses the lower effectiveness of autoencoders in certain attack scenarios by employing a variety of unsupervised algorithms (auxiliary tasks) to supervise the DQN agents. Specifically, we use k-means clustering, autoencoders, isolation forests, and even combinations of these algorithms, such as autoencoder plus isolation forest, to enhance the detection capabilities.

Consider a set of environments $\mathcal{E} = \{E_1, E_2, ..., E_I\}$, where each E_i represents an environment corresponding to a specific type of attack. For each environment E_i, we have: A set of observations $O_i = \{o_{i1}, o_{i2}, ..., o_{iJ}\}$, where o_{ij} represents an observation j in the environment E_i. An unsupervised algorithm T_i is assigned to supervise a DQN agent ($\mathcal{D}_i$) in the environment E_i. Let $\mathcal{S}_i$ denote the state space and $\mathcal{A}_i$ denote the action space of the $\mathcal{D}_i$ respectively. The objective is to learn a policy $\pi_i : \mathcal{S}_i \rightarrow \mathcal{A}_i$ for each $\mathcal{D}_i$, where the policy maximizes the expected cumulative reward $J_i(\pi_i)$ over time:

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \sum_{I=0}^{|\mathcal{E}|} \sum_{j=0}^{|\mathcal{A}|} \gamma^t R^i_{\text{self-supervised}}(s_i^t, a_{i,j}^t) \right]$$

Here, γ is the discount factor, s_i^t is the state at time step t in environment E_i, $a_{ij} = \pi_i(s_i^t)$ is the action selected by the $\mathcal{D}_i$ using policy π in the environment E_i, and $R_{self-supervised}(s_i^t, a_{ij}^t)$ is the reward obtained by doing the a_{ij} in the environment E_i. The training process in our Multi-task Multi-Agent Self-Supervised Approach begins with state representation, where each state s_{ij} suitable for the DQN agent $\mathcal{D}_i$ is derived from the observations O_i. A reward function $R_i : \mathcal{S}_i \times \mathcal{A}_i \rightarrow \mathbb{R}$ is defined based on detection results from the unsupervised algorithm T_i. Each attack-specific agent $\mathcal{D}_i$ is then trained across different environments $\mathcal{E}$ using an RL algorithm to

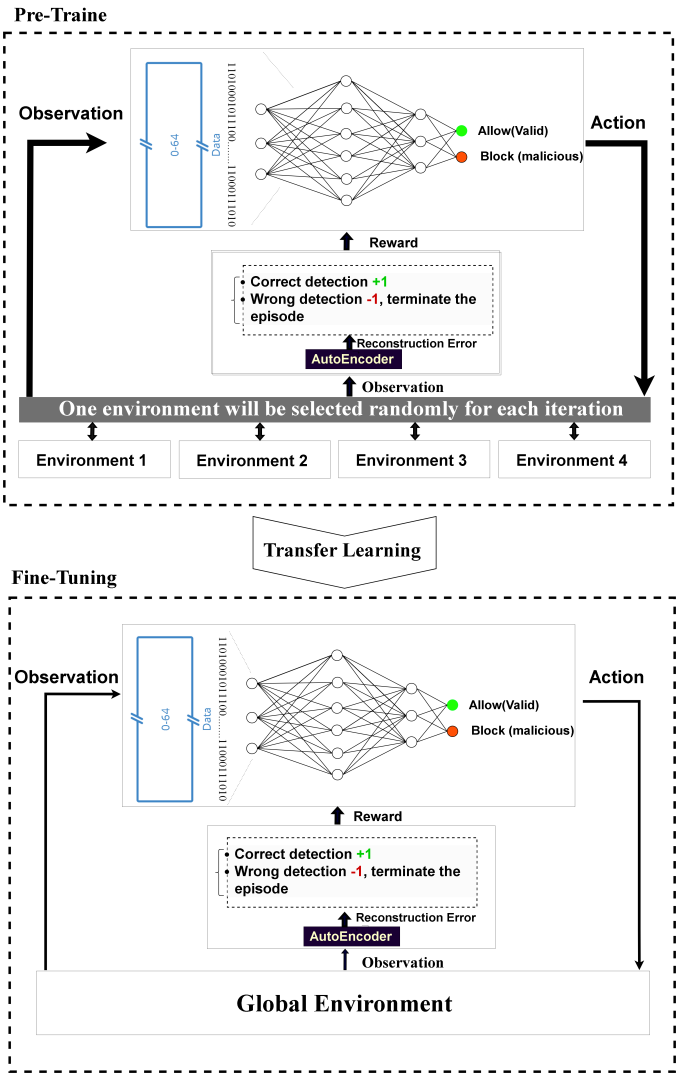

Figure 4: Self-Supervised multi-environment Approach Diagram

maximize the expected cumulative reward $J_i(\pi_i)$. Parameters of the agent's policy π_i are updated iteratively using the Adam optimization algorithm. After specific iterations, a global DQN agent is trained, incorporating predictions from each attack-specific agent. The global agent's reward function is calculated based on these predictions, ensuring a robust and adaptable system for detecting and mitigating cyber-attacks in vehicular networks.

By optimizing the policies π_i for each environment E_i, the Multi-task Multi-Agent Self-Supervised Approach aims to create a robust and adaptable system for detecting cyber-attacks in vehicular networks, capable of effectively leveraging the strengths of different unsupervised algorithms across various attack scenarios. Figure 5 demonstrates the interaction between components in the multi-task multi-agent method, highlighting the training of a global agent.

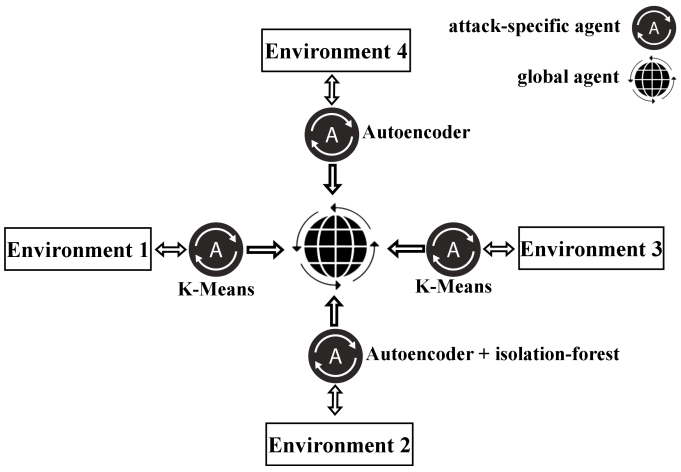

Figure 5: Multi-Task Multi-Agent Approach Diagram

5 Experiment Analysis

We evaluate the performance of our proposed methods in Auto-CIDS, including the *Single-Task Self-Supervised, Multi-Environment Self-Supervised, and Multi-Task Multi-Agent Self-Supervised* approaches.

5.1 System Setup

All experiments were conducted on an HP PC equipped with a 3.30GHz Intel Core i7-7020 processor, 12GB RAM, and a Tesla K80 GPU, running Ubuntu Desktop v18.04 LTS. We implemented our algorithm in Python 3.10 [6], utilizing the Pytorch library version 1.8 [20].

5.2 Dataset

The Car-Hacking dataset trains and evaluates our proposed methods in a simulated environment. This dataset addresses the need to protect in-vehicle networks from cyber-attacks, particularly through message injection. It includes CAN traffic logs from real vehicle attacks via the OBD-II port, featuring DoS, fuzzy, and spoofing attacks on RPM and gear information. Each dataset has 300 intrusion instances, lasting 3 to 5 seconds, totaling 30 to 40 minutes of CAN traffic. Key attributes are **Timestamp** (time in seconds), **CAN ID** (hexadecimal message identifier), **DLC** (data byte count from 0 to 8), **DATA** (data value in bytes, [0 − 7]), and **Flag** (indicating injected (T) or normal (R) message status). This dataset is essential for developing effective automotive cybersecurity defenses.

5.3 Evaluation Metrics

In order to comprehensively evaluate the performance of our proposed methods, we employed a diverse set of evaluation metrics. These metrics include Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$, Precision $= \frac{TP}{TP+FP}$, Recall $= \frac{TP}{TP+FN}$, FNR $= \frac{FN}{FN+TP}$, ER $= \frac{FP+FN}{TP+TN+FP+FN}$, and F1 Score $= 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision}+\text{Recall}}$. Here, TP represents true positives, TN represents true negatives, FP represents false positives, and FN represents false negatives. Given that most previous research in this domain has used only a subset of these metrics, our

approach aims to provide a more meticulous and holistic assessment of the methods' effectiveness. Additionally, we plotted the loss and average reward curves for each method. These plots help visualize the training process and convergence behavior of the models, providing further insights into their learning dynamics and stability. By employing this comprehensive set of evaluation metrics and analyses, we ensure a robust and detailed assessment of our proposed methods, allowing for a clear understanding of their performance in various scenarios.

5.4 Evaluation Result

The evaluation results of the proposed methods are presented and analyzed in the subsequent sections, where we meticulously evaluate these methods and discuss the results in detail. These visualizations offer valuable insights into the effectiveness and robustness of our approaches in the absence of supervision.

5.4.1 Result for Single-Task Self-Supervised Approach. The Single-Task Self-Supervised approach represents our initial effort towards incorporating autonomous capabilities into in-vehicle intrusion detection. The primary objective of this method is to develop an autonomous IDS that operates without human intervention. This experiment evaluates the effectiveness of the proposed method through various performance metrics. Specifically, we conducted experiments using four types of anomalies, including DoS and Fuzzy attacks and spoofing types such as Gear and RPM. We present the results using the loss curve, training accuracy curve, and key evaluation metrics such as precision, recall, accuracy, FNR, ER, and F1-score. These visualizations and metrics provide valuable insights into the model's learning process and its ability to detect attacks autonomously.

Figure 6 represents the training loss observed during the initial training steps (approximately 1,000 iterations) across various datasets. The training process was conducted using a single auxiliary task (autoencoder) to assist the DQN in learning representations from the data.

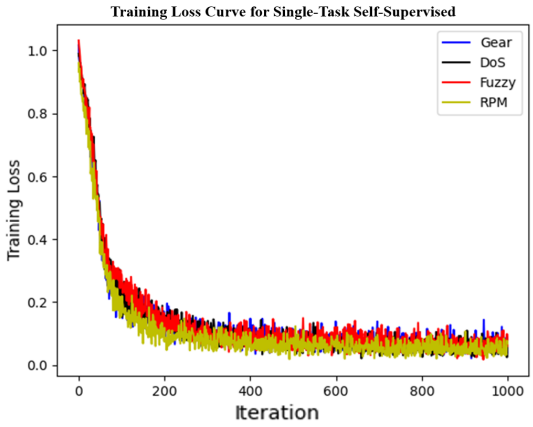

Figure 6: Loss curve for the single-task self-supervised approach

Figure 7: Training accuracy curve for the single-task self-supervised approach

Moreover, Figure 7 illustrates the training accuracy observed during the initial training steps (approximately 1,000 iterations) across various datasets. The training accuracy chart highlights that the accuracy for detecting DoS and RPM attacks is lower compared to other attack types. This discrepancy in accuracy can be attributed to the single auxiliary task (autoencoder) used in the training process. The autoencoder, which assists the DQN in learning representations, struggles to distinguish DoS and RPM attack data due to their high similarity with normal data. As a result, the DQN's performance in accurately classifying these attack types is compromised. This limitation underscores the challenge of detecting certain types of attacks that closely resemble normal traffic, thereby affecting the model's overall accuracy.

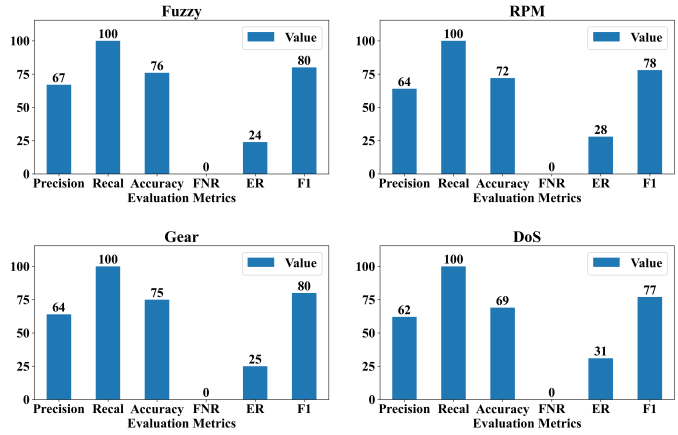

Figure 8: Evaluation metrics for the single-task self-supervised approach

In Figure 8, we present the performance metrics, including precision, recall, accuracy, FNR, ER, and F1-score. These results represent a solid initial step towards developing an autonomous IDS. The findings suggest that the single-task self-supervised approach can identify attacks but requires further refinement to achieve higher

performance levels. This experiment's FNR value equals 0, indicating that the DQN detected all the attacks. However, the ER is relatively high, meaning the DQN agent mistakenly identified some normal data as malicious. This highlights the need for further improvements to reduce false positives while maintaining high detection accuracy. Overall, the initial metrics obtained from this method are promising and provide a foundation for further improvements. The insights gained from this evaluation will guide the enhancements in the subsequent proposed methods.

5.4.2 Result for Self-Supervised Multi-Environment Approach. The primary objective of the Self-Supervised Multi-Environment approach is to develop an autonomous IDS that can generalize learned policies across diverse environments while overcoming the costly process of expert-labeling data. All experimental settings are consistent with those used in the single-task self-supervised method. figure 9 shows the training accuracy and loss curves for the fine-tuning phase of the Self-Supervised Multi-Environment Approach. The curve demonstrates a sharp decrease in loss over the training epochs, indicating that the pre-training phase was effective in transferring knowledge. The training accuracy curve is composed of two distinct curves. The first curve, with the lowest accuracy and represented by a dotted line, depicts the training accuracy during the pre-training phase. The second curve, showing higher accuracy, represents the accuracy during the fine-tuning phase. These results underscore the importance of the pre-training phase in accelerating the overall training process and significantly reducing the need for human intervention.

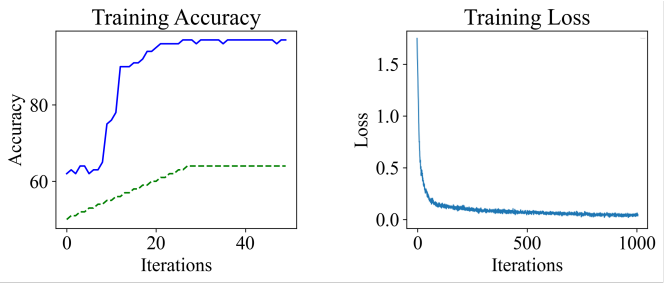

Figure 9: Training accuracy curve for the multi-environment approach on the left side and training loss curve on the right side

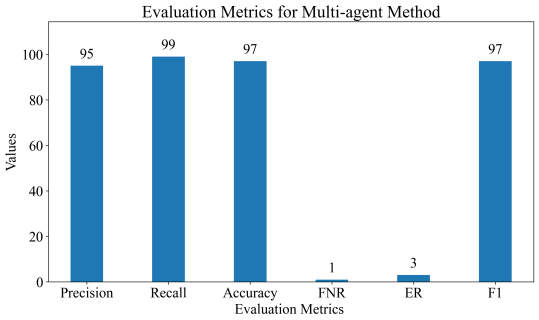

Figure 10: evaluation results for the Self-Supervised Multi-Environment Approach

In addition, Figure 10 presents the evaluation metrics of a self-supervised multi-environment. This approach highlights the capability of a single agent to handle specific attack types, thereby setting the stage for generalization and efficient network traffic monitoring with a unified agent instead of multiple agents for different attacks.

5.4.3 Result for Multi-Task Multi-Agent Self-Supervised Approach. The *Multi-Task Multi-Agent Self-Supervised* Approach's performance has been evaluated using two key diagrams: the training accuracy curve and various performance metrics (precision, recall, accuracy, FNR, ER, and F1-score). This method boasts the highest level of autonomy and has achieved acceptable values for all evaluation metrics.

Figure 11 illustrates the training accuracy and training loss curves for the *multi-task multi-agent self-supervised* approach. The curve shows a significant and consistent increase in accuracy over the training period. This indicates that the multi-agent system effectively learns from diverse tasks and environments, improving accuracy. The high training accuracy suggests that the agents generalize well across different attack types and scenarios. Moreover, figure 12 presents the evaluation metrics for the multi-agent method. The results demonstrate more acceptable and higher values for these metrics than our two previous methods. This improvement highlights the method's enhanced ability to detect and classify various types of attacks autonomously. The high performance across these metrics indicates that the method achieves better generalization and robustness.

The key advantage of this approach is its ability to aggregate the policies obtained by multiple agents in a global agent. This global agent can then interact effectively with the environment, leveraging the combined knowledge and experience from the multi-agent training phase. This results in a highly autonomous and generalized IDS capable of monitoring network traffic and detecting attacks efficiently. Overall, the *multi-task multi-agent self-supervised* approach shows significant promise in advancing the development of autonomous IDS, demonstrating strong performance metrics and effective generalization capabilities.

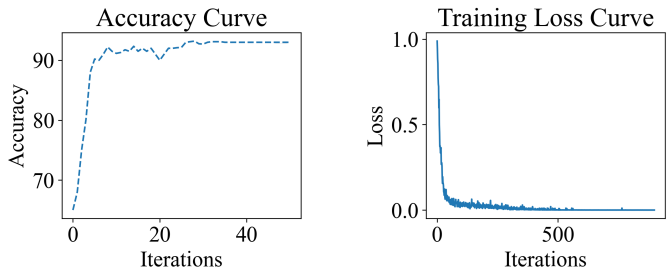

Figure 11: Training accuracy curve for the multi-task multi-agent self-supervised approach on the left side and training loss curve on the right side

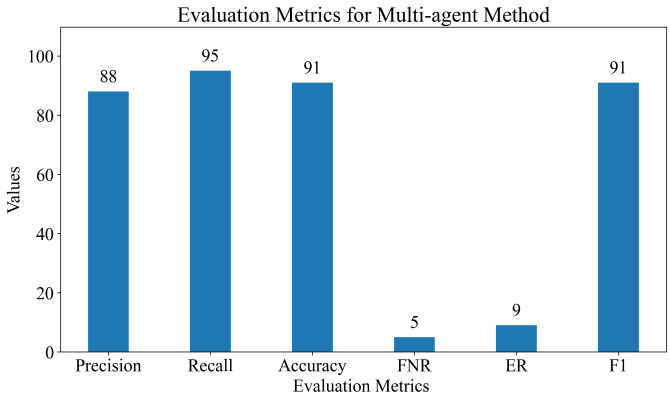

Figure 12: Evaluation results for the multi-task multi-agent self-supervised approach

5.4.4 Comparision Result. To evaluate our proposed methods under unseen conditions, we generated a completely different dataset from the training dataset, with no overlap with the CarHacking dataset. The experiment focused on the multi-environment and multi-agent methods due to their generalization capabilities, as both use a global agent. Figure 13 shows the evaluation results, indicating that the multi-agent method achieves higher accuracy under unseen conditions, which demonstrates the benefits of its multi-task feature that leads to greater generalization as is shown in figure14 with 4% higher for TP metric. Conversely, the multi-environment method, which includes a supervised fine-tuning phase, performs well on similar domains to the training dataset but has lower accuracy in unseen conditions than the multi-agent method.

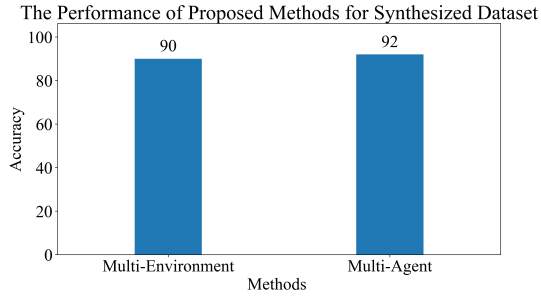

Figure 13: Evaluation results of the proposed methods on an unseen dataset

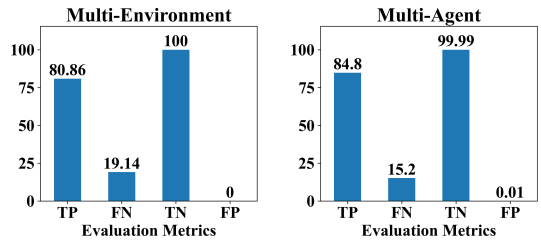

Figure 14: Detailed performance of the proposed methods on an unseen dataset

6 CONCLUSON

This paper presents Auto-CIDS, an autonomous IDS framework with three innovative methods to enhance vehicular network security through self-supervised DRL. Our methods significantly reduce human intervention and improve adaptability across diverse attack scenarios. Comprehensive evaluations demonstrate the efficacy of our approaches in detecting anomalies and cyber-attacks, underscoring the importance of autonomous cybersecurity solutions in modern vehicular networks. These methods have progressively increased the autonomy level of intrusion detection capability of CAN-bus while reducing human intervention and dependence on expert-labeled data while ensuring safety and robustness. The first proposed *Single-Task Self-Supervised* method leverages a pre-trained autoencoder to guide a DQN agent for effective IDS using only normal dataset inputs. The second method, the *Self-Supervised Multi-Environment* method, enhances adaptability across diverse attack scenarios through two-phase training using a normal and minimal labeled dataset. Meanwhile, the *Multi-task Multi-Agent Self-Supervised* method combines various unsupervised algorithms to achieve the highest level of autonomy and effectiveness in intrusion detection across different attack types. Our comprehensive evaluation demonstrates their efficacy in accurately detecting anomalies and cyber attacks in CAN bus systems, underscoring the necessity and effectiveness of autonomous cybersecurity in modern vehicular networks.

References

[1] Easa Alalwany and Imad Mahgoub. 2022. Classification of normal and malicious traffic based on an ensemble of machine learning for a vehicle can-network. *Sensors* 22, 23 (2022), 9195.

[2] Flora Amato, Luigi Coppolino, Francesco Mercaldo, Francesco Moscato, Roberto Nardone, and Antonella Santone. 2021. Can-bus attack detection with deep learning. *IEEE Transactions on Intelligent Transportation Systems* 22, 8 (2021), 5081–5090.

[3] Mehmet Bozdal, Mohammad Samie, Sohaib Aslam, and Ian Jennions. 2020. Evaluation of can bus security challenges. *Sensors* 20, 8 (2020), 2364.

[4] Kun Cheng, Yuebin Bai, Yuan Zhou, Yun Tang, David Sanan, and Yang Liu. 2020. CANeleon: Protecting CAN bus with frame ID chameleon. *IEEE Transactions on Vehicular technology* 69, 7 (2020), 7116–7130.

[5] Abdelouahid Derhab, Mohamed Belaoued, Irfan Mohiuddin, Fajri Kurniawan, and Muhammad Khurram Khan. 2021. Histogram-based intrusion detection and filtering framework for secure and safe in-vehicle networks. *IEEE Transactions on Intelligent Transportation Systems* 23, 3 (2021), 2366–2379.

[6] Python Software Foundation. n.d.. Python 3.10.0 Release. https://www.python.org/downloads/release/python-3100/

[7] Markus Hanselmann, Thilo Strauss, Katharina Dormann, and Holger Ulmer. 2020. CANet: An Unsupervised Intrusion Detection System for High Dimensional CAN Bus Data. *IEEE Access* 8 (2020), 58194–58205. https://doi.org/10.1109/ACCESS.2020.2982544

[8] Thien-Nu Hoang and Daehee Kim. 2024. Supervised contrastive ResNet and transfer learning for the in-vehicle intrusion detection system. *Expert Systems with Applications* 238 (2024), 122181.

[9] Md Delwar Hossain, Hiroyuki Inoue, Hideya Ochiai, Doudou Fall, and Youki Kadobayashi. 2020. An effective in-vehicle CAN bus intrusion detection system using CNN deep learning approach. In *GLOBECOM 2020-2020 IEEE global communications conference*. IEEE, 1–6.

[10] Md Delwar Hossain, Hiroyuki Inoue, Hideya Ochiai, Doudou Fall, and Youki Kadobayashi. 2020. LSTM-based intrusion detection system for in-vehicle can bus communications. *Ieee Access* 8 (2020), 185489–185502.

[11] Camil Jichici, Bogdan Groza, Radu Ragobete, Pal-Stefan Murvay, and Tudor Andreica. 2022. Effective intrusion detection and prevention for the commercial vehicle SAE J1939 CAN bus. *IEEE Transactions on Intelligent Transportation Systems* 23, 10 (2022), 17425–17439.

[12] Shiyi Jin, Jin-Gyun Chung, and Yinan Xu. 2021. Signature-based intrusion detection system (IDS) for in-vehicle CAN bus network. In *2021 IEEE international symposium on circuits and systems (ISCAS)*. IEEE, 1–5.

[13] Vipin Kumar Kukkala, Sooryaa Vignesh Thiruloga, and Sudeep Pasricha. 2020. INDRA: Intrusion Detection Using Recurrent Autoencoders in Automotive Embedded Systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39, 11 (2020), 3698–3710. https://doi.org/10.1109/TCAD.2020.3012749

[14] Brooke Lampe and Weizhi Meng. 2022. IDS for CAN: A practical intrusion detection system for CAN bus security. In *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 1782–1787.

[15] Gabriel Leen and Donal Heffernan. 2002. Expanding automotive electronic systems. *Computer* 35, 1 (2002), 88–93.

[16] Efrat Levy, Asaf Shabtai, Bogdan Groza, Pal-Stefan Murvay, and Yuval Elovici. 2023. CAN-LOC: Spoofing detection and physical intrusion localization on an in-vehicle CAN bus based on deep features of voltage signals. *IEEE Transactions on Information Forensics and Security* (2023).

[17] Alan J Michaels, Venkata Sai Srikar Palukuru, Michael J Fletcher, Chris Henshaw, Steven Williams, Thomas Krauss, James Lawlis, and John J Moore. 2022. CAN bus message authentication via co-channel RF watermark. *IEEE Transactions on Vehicular Technology* 71, 4 (2022), 3670–3686.

[18] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[19] Lotfi ben Othmane, Lalitha Dhulipala, Moataz Abdelkhalek, Nicholas Multari, and Manimaran Govindarasu. 2022. On the Performance of Detecting Injection of Fabricated Messages into the CAN Bus. *IEEE Transactions on Dependable and Secure Computing* 19, 1 (2022), 468–481. https://doi.org/10.1109/TDSC.2020.2990192

[20] PyTorch. n.d.. *Get started with previous versions.* https://pytorch.org/get-started/previous-versions/

[21] Eunbi Seo, Hyun Min Song, and Huy Kang Kim. 2018. GIDS: GAN based intrusion detection system for in-vehicle network. In *2018 16th annual conference on privacy, security and trust (PST)*. IEEE, 1–6.

[22] Hyun Min Song, Ha Rang Kim, and Huy Kang Kim. 2016. Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network. In *2016 international conference on information networking (ICOIN)*. IEEE, 63–68.

[23] Hyun Min Song, Jiyoung Woo, and Huy Kang Kim. 2020. In-vehicle network intrusion detection using deep convolutional neural network. *Vehicular Communications* 21 (2020), 100198.

[24] Heng Sun, Mengsi Sun, Jian Weng, and Zhiquan Liu. 2022. Analysis of ID sequences similarity using DTW in intrusion detection for CAN bus. *IEEE Transactions on Vehicular Technology* 71, 10 (2022), 10426–10441.

[25] Adrian Taylor, Sylvain Leblanc, and Nathalie Japkowicz. 2016. Anomaly detection in automobile control network data with long short-term memory networks. In *2016 IEEE international conference on data science and advanced analytics (DSAA)*. IEEE, 130–139.

[26] Mrinal Thakur, Ahmed Alsinbili, Rahul Chattopadhyay, Elizabeth A Warburton, Kayvan Khadjooi, and Isuru Induruwa. 2024. Identifying the optimal time period for detection of atrial fibrillation after ischaemic stroke and TIA: An updated systematic review and meta-analysis of randomized control trials. *International Journal of Stroke* 19, 5 (2024), 499–505.

[27] Wei Wang, Ming Zhu, Xuewen Zeng, Xiaozhou Ye, and Yiqiang Sheng. 2017. Malware traffic classification using convolutional neural network for representation learning. In *2017 International conference on information networking (ICOIN)*. IEEE, 712–717.

[28] Abbas Yazdinejad, Ali Dehghantanha, Hadis Karimipour, Gautam Srivastava, and Reza M Parizi. 2024. A Robust Privacy-Preserving Federated Learning Model Against Model Poisoning Attacks. *IEEE Transactions on Information Forensics and Security* (2024).

[29] Abbas Yazdinejad, Ali Dehghantanha, Gautam Srivastava, Hadis Karimipour, and Reza M Parizi. 2024. Hybrid privacy preserving federated learning against irregular users in next-generation Internet of Things. *Journal of Systems Architecture* 148 (2024), 103088.

[30] Abbas Yazdinejadna, Reza M. Parizi, Ali Dehghantanha, and Mohammad S. Khan. 2021. A kangaroo-based intrusion detection system on software-defined networks. *Computer Networks* 184 (2021), 107688. https://doi.org/10.1016/j.comnet.2020.107688