INL/JOU-22-67678-Revision-0

RX-ADS: Interpretable Anomaly Detection Using Adversarial ML for Electric Vehicle CAN Data

July 2023

Chathurika Wickramasinghe, Daniel Marino, Harindra Mavikumbure, Victor Cobilean, Timothy David Pennington, Benny J Varghese, Craig G Rieger

Changing the World's Energy Future

INL/JOU-22-67678-Revision-0

RX-ADS: Interpretable Anomaly Detection Using Adversarial ML for Electric Vehicle CAN Data

Chathurika Wickramasinghe, Daniel Marino, Harindra Mavikumbure, Victor Cobilean, Timothy David Pennington, Benny J Varghese, Craig G Rieger

July 2023

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

RX-ADS: Interpretable Anomaly Detection using Adversarial ML for Electric Vehicle CAN data

Chathurika S. Wickramasinghe [1], Daniel L. Marino [1], Harindra S. Mavikumbure [1], Victor Cobilean[1], Timothy D. Pennington[2], Benny J. Varghese[2], Craig Rieger[2], Milos Manic[1] [1] *Virginia Commonwealth University*, Richmond, VA, USA
[2] *Idaho National Laboratory*, Idaho Falls, Idaho, USA
brahmanacsw@vcu.edu, marinodl@vcu.edu, craig.rieger@inl.gov, misko@ieee.org

***Abstract*—Recent year has brought considerable advancements in Electric Vehicles (EVs) and associated infrastructures/communications. Intrusion Detection Systems (IDS) are widely deployed for anomaly detection in such critical infrastructures. This paper presents an Interpretable Anomaly Detection System (RX-ADS) for intrusion detection in CAN protocol communication in EVs. Contributions include: 1) Feature Extractor; 2) Anomaly Detection System; and 3) Explanation Generator for detected anomalies. The presented approach was tested on two benchmark CAN datasets: OTIDS and Car Hacking. The anomaly detection performance of RX-ADS was compared against the state-of-the-art approaches on these datasets: HIDS and GIDS. The RX-ADS approach showed comparable performance to the HIDS approach on OTIDS dataset and outperformed HIDS and GIDS approaches on Car Hacking dataset. Further, the proposed approach was able to generate explanations for detected abnormal behaviors arising from various intrusions. These explanations were later validated by information used by domain experts to detect anomalies. Other advantages of RX-ADS include: 1) the method can be trained on unlabeled data; 2) explanations help experts in understanding anomalies and root course analysis, and also help with AI model debugging and diagnostics, ultimately improving user trust in AI systems.**

***Index Terms*—Neural Networks, Autoencoders, Unsupervised Learning, Anomaly Detection, Explainable AI, Interpretability, Electric Vehicles**

IDS Intrusion Detection Systems
ADS Anomaly Detection Systems
ITS Intelligent Transportation Systems
NN Neural Networks
XAI Explainable Artificial Intelligence
CAN Controller Area Network
HIDS Histogram-based IDS
GIDS GAN-based IDS
DoS Denial-of-Service
OTIDS Offset ratio and Time interval based IDS

I. Introduction

Electric Vehicles (EVs) are becoming a primary component in Intelligent Transportation Systems (ITSs) as it decreases fossil fuel consumption and greenhouse gas emissions, reducing negative environmental impact [1, 2]. In recent years, there has been rapid growth in EV infrastructure, expanding to various areas, including EV manufacturing, charging stations, battery advancements, electric vehicle supply equipment, and other roadside infrastructures [3–6]. Within EV infrastructure, different communication technologies such as vehicle-to-Vehicle (V2V), Vehicle-to-sensor-board (V2S), vehicle-to-infrastructure (V2R), vehicle-to-human (V2H), and vehicle-to-internet (V2I) plays a major role in building resilient operations [7–9]. Security of these technologies is critical to avoid vulnerabilities such as DoS attacks, false data injections, spoofing, and modification [7, 10].

Intrusion Detection Systems (IDSs) are widely used approaches in critical infrastructures such as EV infrastructure [11, 12]. The purpose of IDS is to detect attacks and intruders in communication systems of critical infrastructure, thus avoiding possible catastrophic failures and economic losses. For example, in an EV, attacks can cause break malfunction, engine overheating, control steering issues, and door lock issues, resulting in life-threatening and catastrophic damages [11]. Not only EVs, but other infrastructure components such as charging stations are also prone to severe advanced persistent threats (APT) such as ransomware and malware [12]. Thus building IDSs has become a vital component of EV infrastructure.

During the last decade, data-driven machine learning approaches such as Neural Networks (NNs) have been widely used for building IDSs for various critical infrastructure settings [13, 14]. There are two main types of IDSs: Signature-based IDS and Anomaly Based IDS [15]. Typically, Anomaly Detection systems (ADSs) have the advantage of detecting both known attacks and unknowns/new attacks/abnormalities in the systems [15, 16]. The idea of ADSs is to learn the normal behavior of a system such that anything outside learned normal behavior is detected as an anomaly. The majority of ADSs are trained using only data coming from normal class/behavior, avoiding expensive data labeling process (time-consuming, costly, and requires expertise in data) [17]. Out of widely used NN architectures for ADS development, Autoencoders (AE) have gained much attention due to many advantages such as in-build anomaly detection capability, unsupervised training, scalability, feature extraction, and dimensionality reduction capability. Therefore, in this paper, we are developing a AE-based ADS.

Despite the performance benefits of NNs, people hesitate to trust these systems due to the difficulty of understanding the decision-making process of the AI models, making these

Identify applicable funding agency here. If none, delete this.

systems black-box models [18]. By addressing this, the Trustworthy AI research area has emerged. One main component of Trustworthy AI is the Explainability or Interpretability of AI systems (XAI). XAI aims to provide an understanding of black-box models, enabling users to question and challenge the outcomes of AI systems. It provides many advantages, including justifying outcomes of AI systems, improving trust in AI models, model debugging, and diagnosing [19]. Therefore, this paper presents an explanation generation approach for the presented ADS.

This paper presents the following contributions:

1) **Feature Extractor for CAN bus communication**: We presented a window-based feature engineering approach to extract cyber features from EV communication data. These features are generated to capture the frequency and payload information of the CAN bus communication. We hypothesize that the presented feature extractor is able to generate effective features which can be used by the developed ADS to distinguish normal CAN bus communication from abnormal/attacks behaviors.
2) **Anomaly Detection System implemented using an unsupervised algorithm**: We present an ADS which can be trained only on normal behavior data, without having to use labeled data. Thus, the presented ADS is developed by training ResNet AE using only normal behavior data, and tuning a reconstruction error threshold on the trained normal behavior. We demonstrate that the reconstruction error threshold defined on normal data can be successfully used to identify abnormal communication in CAN bus communication.
3) **Explanation Generation Using Adversarial Machine Learning**: The identified anomalous data are used in our work to generate adversarial samples. These samples are generated by performing the minimum modification required to convert them to the closest normal behavior samples. I.e., adversarial samples are constrained within the distribution of normal behavior data. We create explanations by visualizing the magnitude of the modifications. More specifically, we look at the features that had the largest modifications when generating the adversarial samples. These explanations allowed us to identify important features that help characterize different attacks, allowing us to understand the patterns learned by the models and compare them with expert knowledge.

The presented approach was tested on two benchmark datasets which were provided by the Hacking and Countermeasures Research Laboratory. This approach was developed for an ongoing effort with Idaho National Laboratory (INL) to build ADS for an EV charging system (EVCS). We already deployed some parts of this work in our follow-up paper [20]. Specifically for EVCSs, RX-ADS provides multiple advantages, including unsupervised training, understanding the root courses of a given anomaly, allowing domain experts to distinguish different types of anomalies and common anomaly behaviors, and AI model debugging and diagnostics. Further, to the best of the authors' knowledge, no prior research has been attempted to develop Explainable ADSs for CAN data.

Fig. 1. CAN data frame

The rest of the paper is organized as follows: Section II provides the background and related work; Section III presents the interpretable ADS (RX-ADS); Section IV discusses the experimental setup, results, and discussion, and finally, Section V concluded the paper.

II. Background and Related Work

This section first discusses the Data used for training the presented RX-ADS: CAN, a widely used communication protocol for in-vehicle communication. Then we discuss the current work on IDSs developed using CAN data. Finally, we discuss the background of adversarial machine learning and its applications.

A. CAN Protocol

Controller Area Network (CAN) is the most widely used standard bus protocol for in-vehicle communication. It enables efficient communication between Electronic Control Units (ECUs). It is a broadcast-based protocol that allows multi-master communication, and every node can initiate communication with any other node in the network. Thus, CAN frames do not contain a destination address, unlike other protocols. In CAN protocol, each ECU is able to send messages to the vehicle communication network using data frames [10]. ECUs send frames with their ID number, and the ECU on destination identifies messages by the sender ID included in the frame. The collision of messages and data is avoided by comparing the message ID of the node; the highest priority frame has the lowest ID. CAN is proved to have many advantages, including reducing wiring cost, low weight, low complexity, and operating smoothly in an environment where electromagnetic disturbance factors exist [21].

CAN protocol operates with four main types of frames: the data frame, the remote frame, the error frame, and the overload frame [21]. Most of the communication happens using the CAN data frame. The structure of the CAN data frame is presented in Figure 1, which consists of several common fields that are explained below [21].

- SOF (Start of Frame) - indicates the beginning of the frame.
- Arbitration Field - is composed of message Id and RTR (Remote Transmission Request) bit. Depending on the RTR state the frame will be identified as data or remote frame. During communication frames are prioritized using the ID of the frame.
- Control Field - sends the data size
- Data Field - the actual data that the node wants to send using a data frame, this field can have 0-64 bits.
- CRC Field - contains a 15-bit checksum that is used for error detection

- Ack Field - is used to acknowledge that a valid CAN frame was received by sending a dominant state.
- EOF (End of Frame) - indicates the ending of the frame

B. Anomaly detection using CAN data

Modern vehicles highly rely on ECU communication. Thus CAN has become the standard protocol for facilitating the data exchange between ECUs. While CAN protocol has many advantages, it also consists of security flows such as lack of authentication, vulnerability for various attack vectors, and lack of encryption technologies [22]. In the last couple of years, there has been a surge in research addressing the security and vulnerabilities of CAN protocol, most recent work proposes different IDSs methods. This subsection discusses existing IDS work on CAN data, specifically focusing on neural network methods and feature extraction techniques.

To develop CAN IDSs, there are four types of feature extraction approaches have been tested in the literature [22]. First, *frequency or time-based* features where the timing between CAN frames and sequencing of CAN frame IDs were used for developing CAN IDSs. In [23], the broadcast time interval for each ID within a window (a discrete, non-overlapping, contiguous set of CAN frames) of can frames was calculated. A similar approach has been used in [24], where they calculate the signal co-occurrence time of IDs to calculate the absolute error from expectation for identifying intrusions. The second feature extraction approach is *Payload-based* approach where message content bits are directly used for building CAN IDSs [22]. The third approach is *Signal based* approach where message content is decoded into signal before feeding into the IDS. For example in [25], payload bits are encoded before feeding into Neural network architecture for detecting intrusions in CAN. Finally, the fourth approach is *Physical side channels* where physical attributes such as voltage, and temperature are used to detect intrusions [22, 26]. Other than these four approaches, some IDSs have developed using rule-based methods where the characteristic of CAN communication was encoded into rules for detecting intrusions [22]. The main drawback of rule-based methods is that they require domain knowledge and manual rule generation, which are challenging and time-consuming tasks.

Neural Network (NN) based CAN IDSs are mainly developed by encoding the characteristics of CAN communication into a set of features and training NN algorithms on these extracted sets of features. The features are extracted to ensure capturing the normal behavior patterns of CAN bus communication apart from abnormalities. For example, in [25], Long Short Term Memory (LSTM) and AE-based unsupervised IDS were developed for detecting intrusions. This architecture consists of a neural network architecture where CAN data from each ID type is presented to its assigned LSTM. The results of LSTM networks are aggregated into AE NN. They have tested their approach on Synthetic CAN data only. A similar approach was used in [27] where they used LSTM for detecting anomalies. However, they have not aggregated the results of multiple LSTMs using AE. In [28], deep NN-based IDS was presented, which used deep belief network (DBN) pre-training methods for initial parameter optimization. This approach is a supervised approach where labeled data was required to build the IDS. Convolutional Neural Network (CNN) based supervised CAN IDS was proposed in [29] where they have tested their system on a real CAN data set. They have directly fed information in CAN frames as features for the training of CNN.

C. Adversarial Machine Learning

Adversarial samples are generally referred to as malicious input samples designed to fool machine learning algorithms [30]. These samples are typically created by adding a slight modification into real data samples, such that the outcome of a machine learning model for crafted samples will be different than the real sample [13]. Typically, machine learning models are vulnerable to these generated adversarial samples, resulting in unintended or incorrect outcomes. Generally, adversarial samples are generated to maximize the impact on the model while minimizing the ability to identify the adversarial sample apart from a real sample. This ensures by keeping the adversarial sample inside the domain of valid inputs.

Adversarial machine learning has been widely used for exposing vulnerabilities in critical infrastructures [13]. The positive or negative impact of adversarial ML depends on the purpose of the use of these samples. For example, an attacker can use these samples can be used to gain information on a trained ML model, information on the data set the model trained on, and attack a model. This results in possible privacy invasion, safety failures, data corruption, and model theft [13, 31]. On the other hand, adversarial machine learning also can be used for improving the performance of machine learning models. For example, it can be used to eliminate undefined behaviors of ML models, exploit vulnerabilities, assess model robustness and improve generalization [13, 31–33]. This paper uses adversarial ML to interpret CAN ADS, which helps with interpreting the decision-making process of black-box NN models and helps with NN model debugging and diagnostics.

III. RX-ADS Methodology

This section discusses the development of RX-ADS. This section consists of three subsections. First discusses the system model, articulating various aspects of the systems as presented in Figure 2. The second section discusses the main components of the system in detail. The third section discusses the assumptions, challenges, and limitations of the system.

A. System Model

The proposed system is illustrated in Figure 2. The proposed RX-ADS consists of several main components: feature extractor, CAN bus data, Machine Learning (ML) model, Adversarial Approach, and Interpretable Interface. First, we extract a set of cyber features from raw CAN communication data using the feature extractor. It converts the raw CAN data into a data format that can feed into ML algorithms. This data can be collected from CAN bus communication within EV vehicles or between EV and the EV chargers as shown

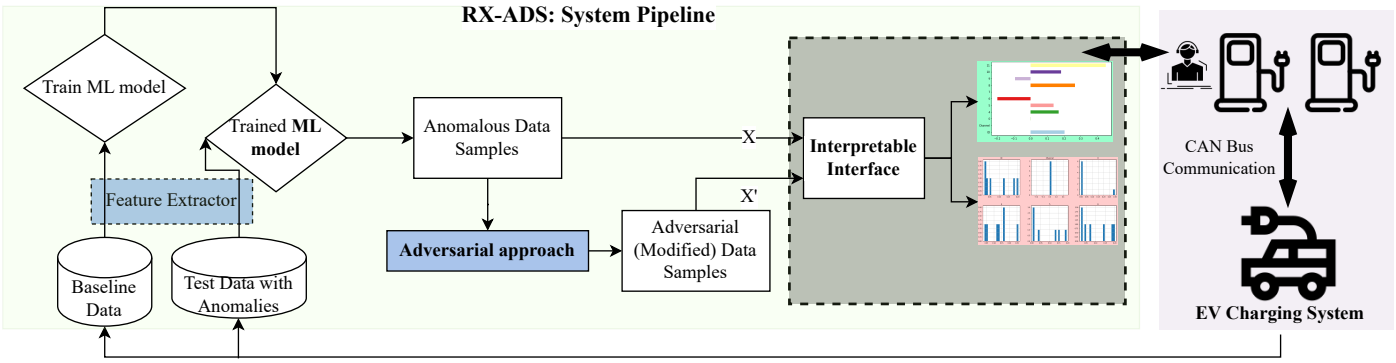


Fig. 2. RX-ADS: Interpretable Anomaly Detection System Framework

in Figure 2. Then, it trains an ML model using baseline data, i.e., data that represent the normal behavior of the system. Adversarial ML is used for generating the explanation for abnormal behaviors. Once it identifies abnormal samples, these samples are modified using an adversarial approach, i.e., it performs the minimum modification required to change the anomalous records (x') into normal/baseline records (x). The difference between x and x' is used to in the Interpretable Interface for illustrating the most relevant features that lead to anomalous behaviors.

B. Main Components

1) Feature Extractor/ Feature Engineering: This subsection describes the window-based feature engineering approach where we generate a set of cyber features from raw CAN frames. This is motivated by widely used window-based network flow feature extraction methods in industrial control ADSs [34]. The main goal of this approach is to extract a set of features using a set of CAN messages within a defined-sized time window. These features are selected based on the available literature on CAN bus data [10, 35]. The sliding window-based feature extraction algorithm is presented in Algorithm I in Table I. For each dataset, we used a time window and extracted features on CAN frames within that window. Overlaps between two windows are kept at half of the window size. In this experiment, window features are extracted using different time window sizes ($winSize$).

The set of extracted features with their description is presented in Table II. These features are extracted to represent the fluctuation in normal behavior in the CAN communication data. All or some of the features are extracted for each tested dataset. Then, the extracted features were fed into the AE model for building the ADS.

2) Machine Learning Model: As discussed in the introduction, Autoencoder (AE) NN model was used to learn the normal/baseline behavior of the system. Specifically, we used deep ResNet AE (RAE) architecture to avoid possible performance degradation, and easy parameter optimization [17]. AE has an encoder and decoder, each consisting of multiple hidden layers. Training of the model consists of two stages, the encoding stage, and the decoding stage. The encoding stage transforms the input data into an embedded representation, whereas in the decoding stage, the embedded representation is reproduced back to the original input record (reconstruction). Encoding and decoding functions are non-linear transformation functions; in this experiment, we used a Sigmoid function. The loss function (J_θ) of the AE model is computed using the difference between the input (x) and the reconstruction (x'). Thus reconstruction error of the AE is calculated as follows,

$$J_\theta = \frac{1}{T}\sum_{i=1}^{T} \|x_i - x_i'\|^2 \tag{1}$$

where x_i is the i th input sample, x_i' is the reconstruction for ith input sample, θ denotes the set of parameters of the AE (weights and biases).

During training, AE is trained with data coming from the normal behavior of the system. Therefore, it only learns the possible normal behaviors of the system. When unseen records are presented to the trained AE, the amount of reconstruction error indicates how much the presented data differs from the learned normal behavior. A threshold value is defined to identify possible anomalies. The data records were detected as anomalies if the reconstruction error was higher than the defined threshold value. Thus, given data record x_i is detected as anomaly ($y = 1$) or normal ($y = 0$) as follows,

$$J_{\theta,i} = \|x_i - x_i'\|^2 \tag{2}$$

$$y = \begin{cases} 1 & : J_{\theta,i} \geq th \\ 0 & : J_{\theta,i} < th \end{cases} \tag{3}$$

where $J_{\theta,i}$ is the reconstruction error of ith data record, th denotes the threshold value, and y represents predicted label: anomaly or not. The threshold value is optimized based on the training baseline data, i.e., the threshold value should capture the baseline data boundary, capturing the normal behavior fluctuations.

L1 regularized RAE architecture was used as some of the features can result in data sparsity. The mean squared error was used as the loss function. A different number of hidden layer sizes were tested, and the paper presents the best results. We divided the baseline/normal data into two sets (train/test) with a 0.7/0.3 ratio. The data was scaled to the 0-1 range. Once the RAE model is trained with baseline data, the reconstruction

TABLE I
PRESENTED FEATURE EXTRACTION METHOD

Algorithm I: Extract features	
Inputs: Dataset (X), Time window size ($winSize$), Possible set of signal IDs ($IDList$)	
Outputs: Window features	
1:	$startTime = 0$
2:	$listRecords = []$ ← = Initialize a list to store window features
3:	$endTime = Timestamp of last record of X$ ← = Store the last timestamp of the dataset
4:	% Calculating features for each overlapping time window
5:	**while** $startTime < endTime$ do
6:	$windowMessages$ ← Extract massages from X where timestamp is within range $startTime - (startTime + winSize)$
7:	$no\_of\_records$ ← Number of messages in $windowMessages$
8:	no_of_ids ← Number of unique IDs in $windowMessages$
9:	no_of_dlc ← Number of unique data length of messages in $windowMessages$
10:	$time\_interval$ ← Minimum/Maximum/Mean timestamp differences of messages in $windowMessages$
11:	$no\_of\_req\_msgs$ ← Number of remote frames in $windowMessages$
12:	no_of_res ← Number of response frames in $windowMessages$
13:	no_of_lost ← Number of lost response frames in $windowMessages$
14:	$ratio$ ← Number of messages between requests and responses in $windowMessages$ (Minimum, Maximum, Mean)
15:	$reply\_time\_interval$ ← Minimum/Maximum/Mean timestamp differences of requests and responses in $windowMessages$
16:	0000 ← Number of high priority messages (ID=0000) in $windowMessages$
17:	no_XXXX ← Number of messages with $ID = XXXX$ in $windowMessages$
18:	$payload\_pX\_XXXX$ ← Mean signal values of payload signal x from messages with $ID = XXXX$ in $windowMessages$
19:	$startTime+ = (winSize/2)$ ← Calculate start time of next window
20:	**end while**

TABLE II
FEATURE LIST

Feature	**Description**
no_of_records	Number of CAN messages
no_of_ids	Number of unique CAN message IDs
no_of_dlc	Number of unique CAN message payload lengths
time_interval	Time interval between messages(two consecutive messages: minimum, maximum and mean)
no_of_req_msgs	Number of request frames
no_of_res	Number of responses frames
no_of_lost	Number of lost response frames
ratio (min, max, mean)	Number of messages between request frame and response frame
instant_reply_count	Number of instant reply messages
reply_time_interval (min, max, mean)	Time difference between request frame and corresponding response frame
high_priority_count/ 0000	Number of high priority messages
no_XXXX	Number of messages with ID XXXX
payload_P1_XXXX	Mean payload of signal P1 with ID XXXX

errors on train data were used to define an error threshold by keeping 99.9% of train data within the defined threshold. The trained RAE's (ADS) performance was evaluated using the test baseline data and abnormal/attack data.

3) Adversarial ML: Modifying Anomalous Samples: This paper uses adversarial Machine Learning (ML) to understand why a given sample is detected as an anomaly. The concept of adversarial sample generation was used to find the minimum modification needed to change the anomalous sample into a normal behavior sample. This is achieved by finding an adversarial sample x'' that is detected as a normal sample with the given th while minimizing the distance between real sample x and adversarial/modified sample x''.

$$\min_{x''} \|x' - x''\|^2 \tag{4}$$

$$\begin{aligned} s.t : J_{\theta,x''} &\leq th \\ x_{min} <= x'' <= x_{max} & \end{aligned} \tag{5}$$

We constrain the adversarial sample x'' should be inside the bounds (x_{min}, x_{max}). These bounds are defined using the training data, ensuring that the adversarial samples are inside the domain of data distribution. In the presented approach, this magnitude of modification is considered to be proportional to the feature contribution toward identified attacks. These required modifications are used in Interpretable Interface to generate explanations for identified attacks.

4) Interpretable Interface: The presented interpretable interface generates explanations for detected anomalies. For generating explanations, RX-ADS uses the identified anomalous samples as references, then uses Eq 4 and 5 to find the adversarial samples with minimum modifications. Explanation generated under two categories:

- Explanations for individual Anomaly Samples: These explanations are generated by calculating the difference between the anomaly sample and the closest adversarial sample ($x - x''$) and visualizing it using a bar chart. This bar chart shows the deviation of the anomaly sample from what the model learned as normal behavior. Domain experts can analyze these graphs quantitatively to understand the root courses of a given anomaly. For example, identifying cyber anomalous behavior which leads to a physical impact in the vehicle.
- Explanations for global anomalous behavior: Explanations generated for anomalous data records are aggregated to understand common anomaly behaviors in the system. It allows us to distinguish different types of anomalies and their common behaviors.

C. Assumptions, Challenges, and Limitations

There are a couple of assumptions, challenges, and limitations associated with the proposed approach. To train this approach, it is essential to have a large enough data set, representing the normal behavior of the system. Thus, the training data set is assumed to cover all possible representations of normal communication behaviors of the system. The model

must be re-trained in the case when there are new behaviors that are significantly different from trained normal behaviors. Otherwise, these behaviors will be detected as anomalies, increasing false positives. Since CAN bus standards can be different from one vehicle/system to another, the transferability of the system can be limited. If the new behavior contains different CAN IDs, new IDs should be integrated into the feature-extraction methods when re-fitting the system. Finally, when an anomaly is associated with a large number of features, then the Adversarial Approach might reveal many deviated features. Thus, effectively presenting this information to domain experts in a real-time setup can be challenging. In this work, it is assumed that anomaly communication happens only on a limited number of features, allowing us to present them clearly in a GUI using a bar plot. We are currently working on deploying this system in an ongoing effort with Idaho National Laboratory (INL) to an EV charging system (EVCS). We are planning on exploring options for addressing these challenges.

IV. Experiment, Results, and Discussion

This section first discusses the rationale for selected datasets, comparative analysis approaches, and validation methods. Then it presents results and discussions for each dataset.

The objective of the evaluation is two-prong, 1) to achieve comparable or improved anomaly detection performance compared to the state-of-the-art methods on the experimented datasets, 2) to identify features that have the highest relevance towards identified attacks and to validate them through domain experts. The proposed system was tested against two popular CAN protocol benchmark datasets: OTIDS and Car Hacking. These are two publicly available EV CAN protocol datasets in the recent literature. Both of these datasets are released by the Hacking and Countermeasures Research Lab (HCRL) for the research community on CAN bus communication. They contain CAN bus data representing normal behavior and abnormal/attack behaviors, allowing us to use them for building our anomaly detection system. The Car Hacking dataset is the most recent dataset released by them. OTIDS is the only publicly available dataset with remote frames and responses. Based on the domain knowledge of CAN data, loss of remote frames is a major indication during DoS attacks. Therefore, it is essential to feed this information as features into the algorithms and check whether the proposed adversarial approach outputs them during DoS attacks. Since these datasets are used for the development of IDSs, domain knowledge is available on different attacks, providing the main CAN communication characteristics for each attack type. This information allows us to validate the RX-ADS explanation by checking whether the features identified by the adversarial approach are discussed in the literature for detecting these attacks.

We selected state-of-the-art performing methods found in the literature which include the Histogram-based Intrusion Detection System and the GAN Intrusion Detection System (HIDS and GIDS). We used the same key performance indicators implemented in them including accuracy, precision, recall, and F1. Further research will be conducted in future work to further validate these explanations once deployed in a human-in-loop setup with our domain expert partners from INL. Dataset-specific information is as follows:

TABLE III
OTIDS dataset: RX-ADS anomaly detection comparison with recent literature

Approach	Normal Behavior	DoS	Fuzzy
HIDS [35]	100%	100%	100%
OCSVM	99.77%	100%	100%
LOF	99.32%	100%	100%
RX-ADS	**100%**	**100%**	**100%**

OTIDS: The OTIDS benchmark CAN dataset was released by Hacking and Countermeasures Research Lab (HCRL) [21]. It contains real CAN data collected from a Kia Soul vehicle in normal behavior and during a set of attacks: DoS, Fuzzy, and impersonate. Currently, HIDS is the state-of-the-art performing method on this dataset [35]. It has to be noticed that this is the only open dataset with remote frames and responses, such that it is important to experiment on how this information is essential for anomaly identification.

Car Hacking: The Car Hacking dataset is the most recent dataset released by the Hacking and Countermeasures Research Lab (HCRL). This dataset contains real CAN data collected from Hyundai YF Sonata. It contains a baseline data file, a data file with DoS attacks, a data file with Fuzzy attacks, and a data file with Spoofing attacks. CAN frame structure is very similar compared to their first released OTIDS dataset. This dataset seems to be the most widely used dataset in the CAN IDS literature [22]. The Generative neural network-based IDS (GIDS) is the state-of-the-art approach for detecting intrusions in the Car Hacking dataset [10].

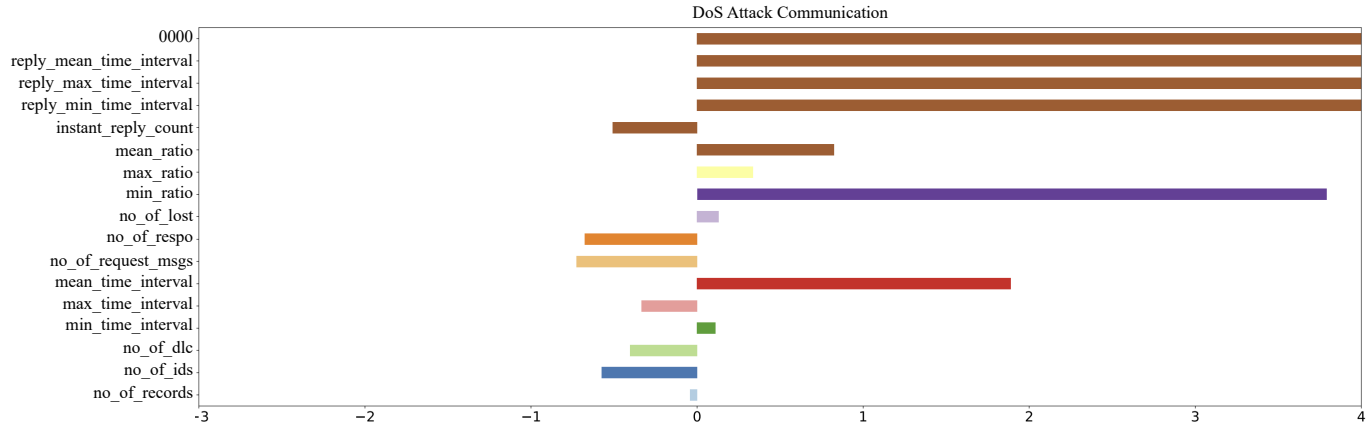
The rest of the section discusses RX-ADS results and discussion for each dataset.

A. RX-ADS vs. HIDS (OTIDS dataset)

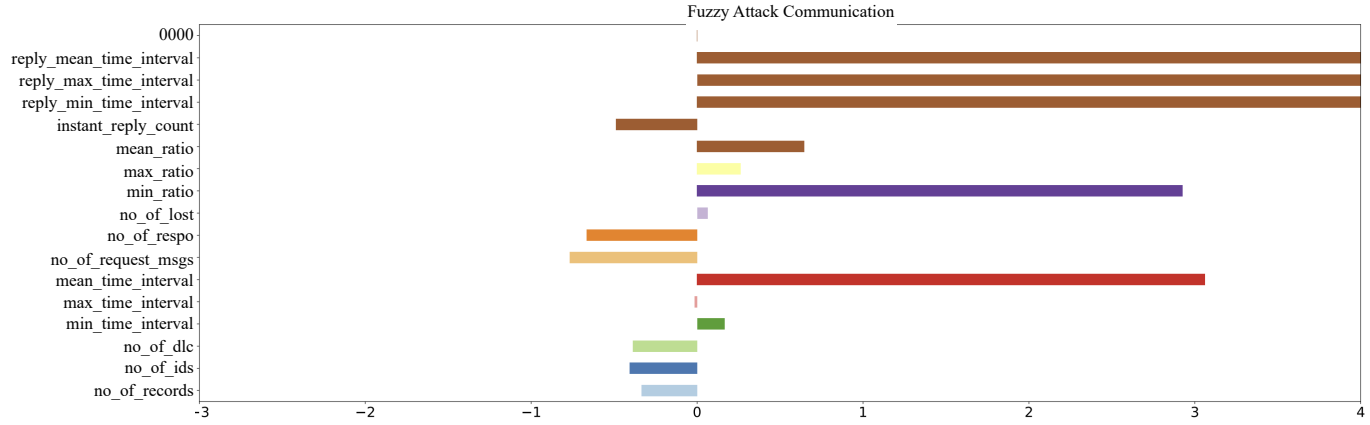
This section compares the anomaly detection performance between RX-ADS and HIDS algorithms.

Data Processing: This experiment used baseline, DoS, and Fuzzy data to simplify the experiment and compare the explanations. DoS Attacks have been implemented by injecting messages of '0x000' CAN ID in a short cycle. Fuzzy Attacks have been implemented by injecting messages of spoofed random CAN ID and DATA values. All the features except $payload\_PX\_XXXX$ were extracted for this dataset. This was performed due to the available domain knowledge on this dataset shows that it is possible to identify abnormalities by using only remote request and response-based features. However, this feature showed its value in our EV application of this approach in [20].

1) Anomaly Detection System Performance: We experimented with different millisecond time window sizes:0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5. We found that when the time window is too small ($<$0.02), the performance of the fuzzy attack detection rate decreases. Further, the baseline accuracy was reduced if the time window is too large (0.1$>$).



(a) DoS



(b) Fuzzy

Fig. 3. OTIDS dataset: Outcomes generated using the adversarial approach for DoS records and Fuzzy records

TABLE IV
OTIDS DATASET: NATURAL INTERPRETATION OF ABNORMAL COMMUNICATION COMPARED TO NORMAL

Characteristics of communication	**Normal communication**	**Abnormal communication (DoS and Fuzzy)**
High priority CAN frames with ID 0000	Low	High
Min/max/mean time interval between remote and response messages	Low	High
Number of CAN messages between the remote frame and its corresponding response frame (min/max/mean ratio)	Low	High
Higher number of lost response messages	Low	High
Number of instant reply, request, and response messages	High	Low
Number of unique IDs, number of records, and unique DLC values	High	Low

The best-observed result was observed for 0.05 milliseconds window size, which is recorded in this section. Table III shows the detection performance of presented RX-ADS compared to recent state-of-the-art IDSs on OTIDS dataset: Histogram-based approach (HIDS) presented in [35]. We also implement two widely used anomaly detection algorithms: Local Outlier Factor (LOF) and One-Class SVM (OCSVM).

It can be seen that RX-ADS shows comparable performance with the state-of-the-art approach on this dataset. Both HIDS and RX-ADS used window-based feature exaction methods. The HIDS uses a fixed number of CAN frames as a window, whereas RX-ADS uses CAN messages within a fixed time window. It has to be noted that the HIDS uses the K-Nearest Neighbor (KNN) algorithm for performing multi-class classification. Thus it requires labeled data from all the classes (normal, DoS, and Fuzzy) for training. However, RX-ADS only requires data from normal behavior for training, which is advantageous as the data labeling is expensive [17, 19].

2) Explanation generation: Figure 3 shows the deviations calculated using the adversarial method for two types of abnormal/attack behaviors (DoS and Fuzzy). As we discussed before, these deviations of feature values not only explain the behavior of attacks compared to normal behavior but also help with distinguishing different types of attacks. To make the comparison easy, deviations for two types of attacks were presented with the same scale. Explanations for attack behaviors can be naturally interpreted as compared to normal behavior in the following tabular format in Table IV.

Related literature on this dataset confirms that normal communication has a very low lost reply rate, a higher number of instant reply rates, and a very low/zero amount of high-priority messages. Further, a higher number of messages during normal communication results in a higher number of unique IDs, number of records (messages), and unique DLC values. However, during attacks, it generates high-priority messages or spoofs random messages, resulting in collisions between CAN frames. It leads to a delay in CAN communication. Thus, fewer records are expected during abnormal communication compared to baseline. Further, this results in a fewer number of unique DLCs, IDs, and a number of records within a window. Thus, the identified features of attacks match the domain expert's knowledge of this dataset.

The explanation generated for two types of attacks can be compared against each other to identify distinguishing characteristics between them. The explanations for distinguishing two behaviors can be naturally interpreted in the following manner.

DoS and Fuzzy attacks affect the system differently based on the following observations:

- DoS results in a higher number of high-priority messages (0000), whereas Fuzzy does not result in high-priority messages with ID 0000
- Min/Max/Mean ratio is higher for DoS due to high-priority message communication.
- No of lost response frame rate is higher for DoS.

Once adversarial samples are generated, feature value distribution of baseline, attacks, and adversarial records also give insights into how different features behave under abnormalities. Figure 4 illustrates the feature behavior for selected features under DoS attacks. It can be seen that many of the identified features deviate from the baseline behavior with different magnitudes (Orange line). However, generated adversarial samples (green line) have a much closer feature value distribution to the baseline (blue line). Feature value distribution during Fuzzy attacks is also presented in Figure 5, which also shows similar behavior.

B. RX-ADS vs. GIDS and HIDS (Car Hacking dataset)

This section compares the anomaly detection performance between RX-ADS with GIDS and HIDS algorithms.

Data Processing: In this experiment, we used only baseline, DoS and Fuzzy CAN data. All the features except payload PX_XXXX were extracted for this dataset. This was performed due to the available domain knowledge confirming that it is possible to identify intrusions only using timing information and ID frequencies. Remote frame indicators are only included in baseline data; thus, this experiment ignores remote information bits from CAN frames. Initial data analysis indicated large gaps between CAN frames during attacks [22, 36]. Our analysis also confirmed this behavior. Hence we trimmed attack datasets before using them for experimentation.

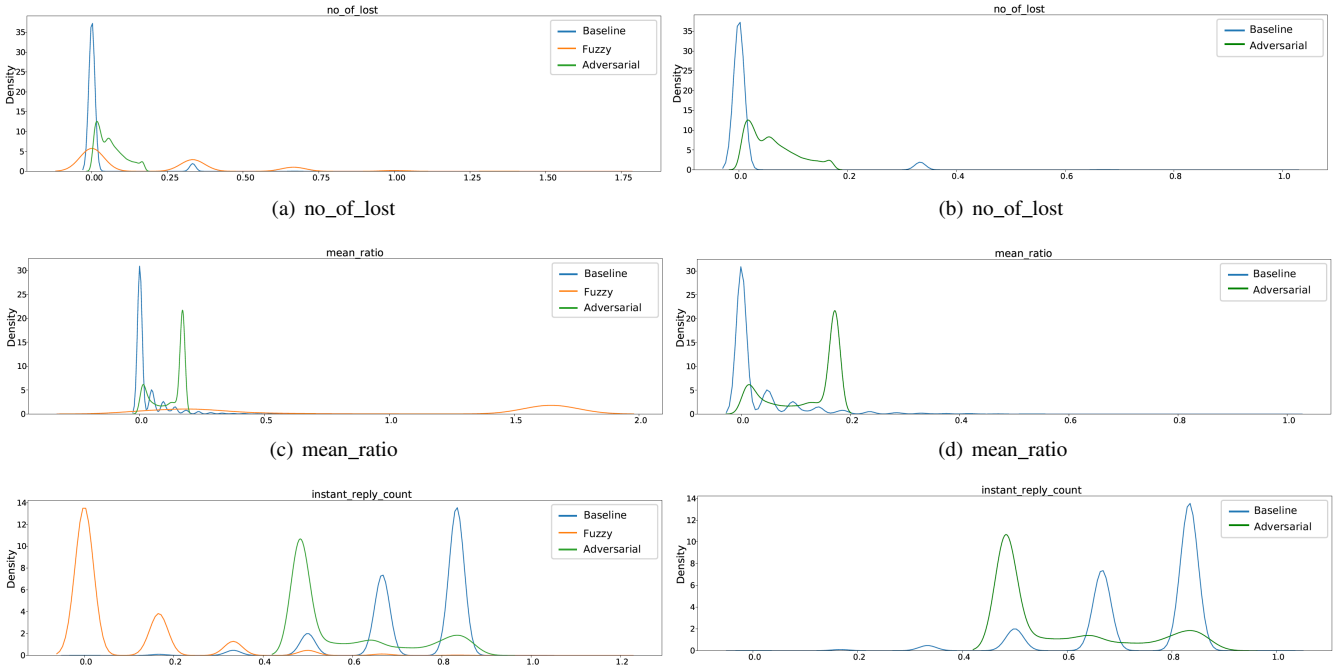
1) Anomaly Detection System Performance: We experimented with different millisecond time windows: 0.01, 0.02, 0.03, 0.04, 0.05. When looking at frames within the window, we observed some differences between normal and attack communication. When the window size is too small, there are many windows without injected intrusion frames (During attack communication). However, even without any injected frames, the window features of CAN frames are different due to the fact that this communication happens during attacks. Thus, considering these windows as normal windows is inaccurate. If the time window is too large, the number of records generated from window-based feature extraction decreases. This results in less number of data records for training. Out of the tested time windows, 0.03 and 0.04 milliseconds gave the best results. The best-observed results were recorded in this section. Table V shows the detection performance of presented RX-ADS compared to recent state-of-the-art IDSs on Car hacking dataset: Histogram-based approach (HIDA) presented in [35] and GIDS presented in [10]. Further, we also compared these results with two classic anomaly detection algorithms: OCSVM and LOF. We calculate Accuracy, precision, recall, and F1 scores for comparison purposes with available literature.

TABLE V
CAR HACKING DATASET: RX-ADS ANOMALY DETECTION COMPARISON WITH RECENT LITERATURE

Method	Data	Accuracy	Precision	Recall	F1
HIDS	*DoS*	97.28	100	96.2	98.06
	Fuzzy	95.17	99.55	94.3	97.18
GIDS [10]	*DoS*	97.9	96.8	99.6	95.42
	Fuzzy	98.0	97.3	99.5	98.39
OCSVM	*Baseline Test*	99.53	-	-	-
	DoS	51.12	51.10	100	67.64
	Fuzzy	64.74	60.82	99.99	75.63
LOF	*Baseline Test*	99.67	-	-	-
	DoS	96.25	93.21	99.94	96.46
	Fuzzy	97.44	95.57	99.96	97.71
RX-ADS	*Baseline Test*	100	-	-	-
	DoS	99.47	99.6	99.74	**99.67**
	Fuzzy	99.19	99.39	99.63	**99.51**

It can be seen that the anomaly detection rate of RX-ADS is higher for both DoS and Fuzzy intrusions compared to other approaches. RX-ADS showed the highest F1 score whereas OCSVM showed the lowest F1 score. As we discussed before, RX-ADS has an advantage over the HIDS approach as RX-ADS does not require labeled data for training. Further, HIDS has implemented different variants of OCSVM-attack models for each intrusion, whereas RX-ADS only implements one model. RX-ADS can use aggregated explanations for distinguishing DoS from Fuzzy intrusions. GIDS is similar to RX-ADS as they only train on normal data. GIDS requires converting CAN data into image format for training [10]. Thus it is a complex and expensive pre-processing step compared to the simple window-based feature extraction used in RX-ADS. Another advantage of RX-ADS is the model interpretability, making domain experts verify model outcomes.

2) Explanation generation: Figure 6 shows the deviations calculated using the adversarial approach for two types of abnormalities (DoS and Fuzzy). It can be seen that some features highly deviated during abnormal behaviors compared



(a) no_of_lost (b) no_of_lost

(c) mean_ratio (d) mean_ratio

(e) instant_reply_count (f) instant_reply_count

Fig. 4. DoS: Feature value distribution of DoS data and Adversarial data compared to normal feature value distribution

TABLE VI
CAR HACKING DATASET: NATURAL INTERPRETATION OF ABNORMAL COMMUNICATION COMPARED TO NORMAL

Characteristics of communication	**Normal communication**	**Abnormal communication**	
		DoS	**Fuzzy**
High priority CAN frames with ID 0000	Low	High	Low
Odd ID can frames within a window	Low	Highest	High
Min/max/mean time interval between remote and response messages	Low	High	Highest
Number of unique CAN frames within a window	Low	Low	High

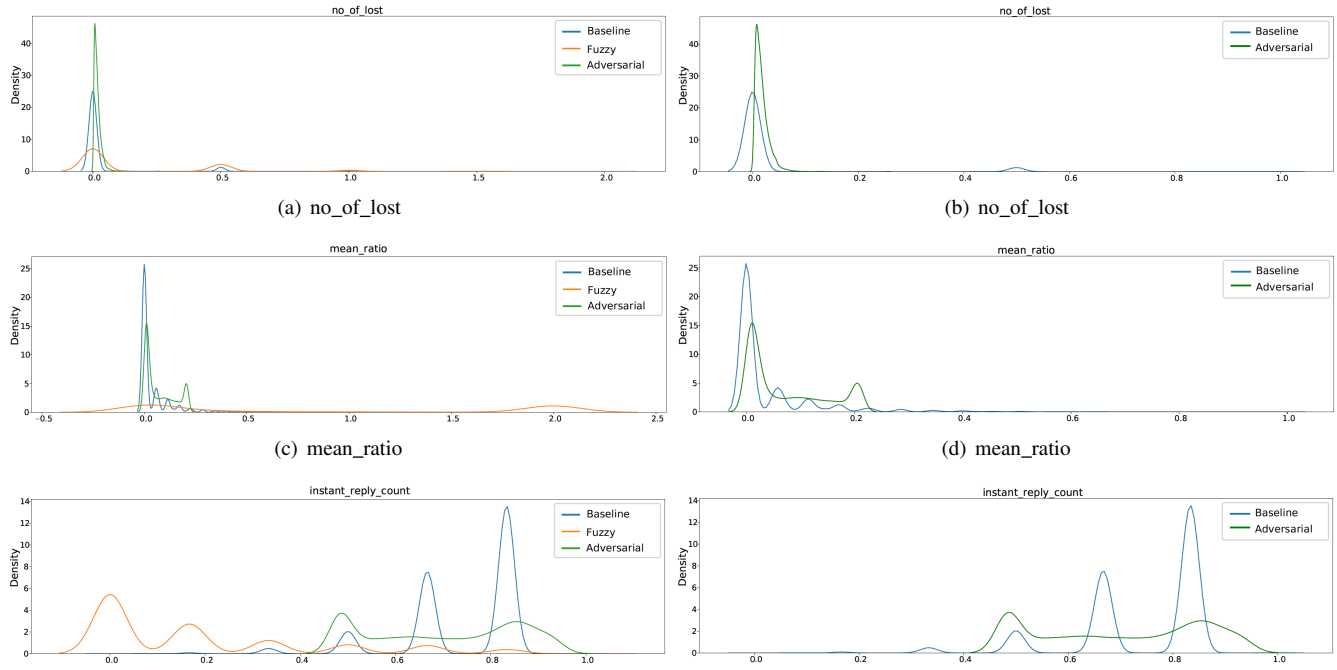
to baseline. Further, there is a clear difference between the two abnormal behaviors. Explanations for attack behaviors can be naturally interpreted in the following tabular format in Table VI.

Compared to the baseline, there are more frames with ID 0000 and odd IDs during DoS attacks. Further, the mean time interval between frames is higher during DoS attacks. Compared to the baseline, a higher number of unique ID frames can be seen during Fuzzy attacks.

We can also compare the behaviors between two attacks for distinguishing characteristics. During attack behaviors, both Fuzzy and DoS show a higher number of odd ID frames (frames with IDs that have not been encountered during baseline behavior). However, during DoS, these are mainly coming from frames with ID 0000, whereas in Fuzzy, these are not high-priority frames. These are coming from random IDs which haven't encounter during baseline behavior. The mean time interval between frames is higher for both compared to the baseline. However, the Fuzzy attack shows the highest mean time interval than DoS. The number of unique IDs is very high during fuzzy attacks compared to DoS.

Related literature on this dataset confirms the above-discussed behavior. For example, normal communication has a very low number of high-priority messages. In addition, the Fuzzy attacks result in CAN frames with random IDs which have not been encountered during baseline behaviors. During DoS, the number of high-priority messages with ID 0000 is higher compared to baseline and Fuzzy attacks. Both attacks result in fewer frames within a time window. This happens because, during attacks, it generates high-priority messages or spoofs random messages. These high-priority frames and other attack frames can have multiple effects, such as packet collisions and paralyzing the functions of a vehicle resulting in delays or even suspension of other messages [21]. Therefore, the mean time interval between frames is higher compared to the baseline. These may be due to the CAN frame collisions and paralyzing the functions of a vehicle resulting in delays, or even suspension of other CAN messages [21].

Figure 7 shows the explanations generated for normal communication frames, which are in-between attack behaviors. Even though these samples do not have injected attack frames, the overall communication pattern of these windows was significantly different from the baseline. Thus, the error threshold value was increased to detect these windows as normal windows. This can also be used as a similar filtering method to detect windows without injected frames within attacks (a similar filtering approach was proposed in HIDS). These normal windows during attack communication are expected to

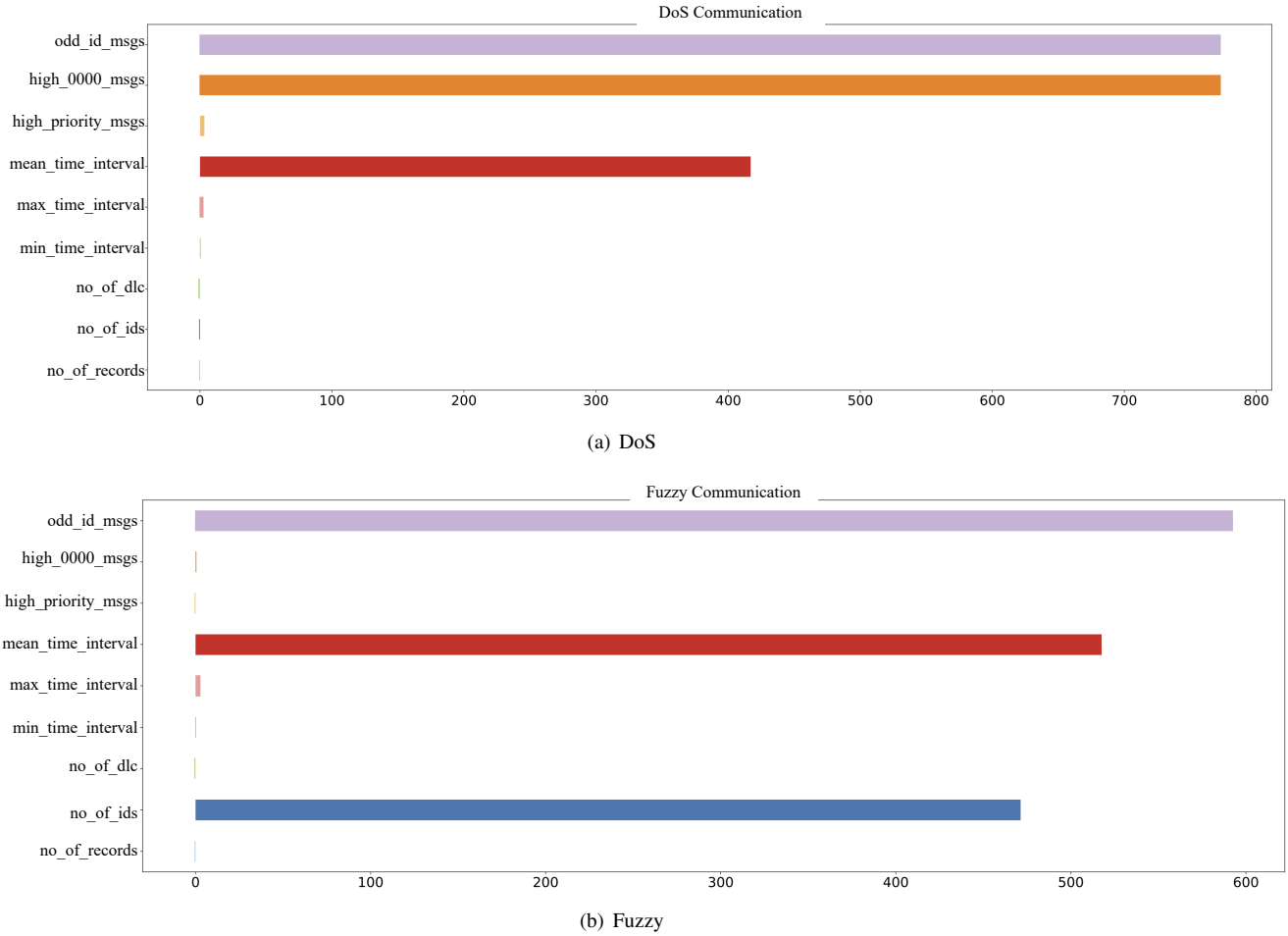


(a) no_of_lost (b) no_of_lost

(c) mean_ratio (d) mean_ratio

(e) instant_reply_count (f) instant_reply_count

Fig. 5. Fuzzy: Feature value distribution of Fuzzy data and Adversarial data compared to normal feature value distribution



(a) DoS

(b) Fuzzy

Fig. 6. Car Hacking Dataset: Outcomes generated using adversarial approach for DoS records and Fuzzy records

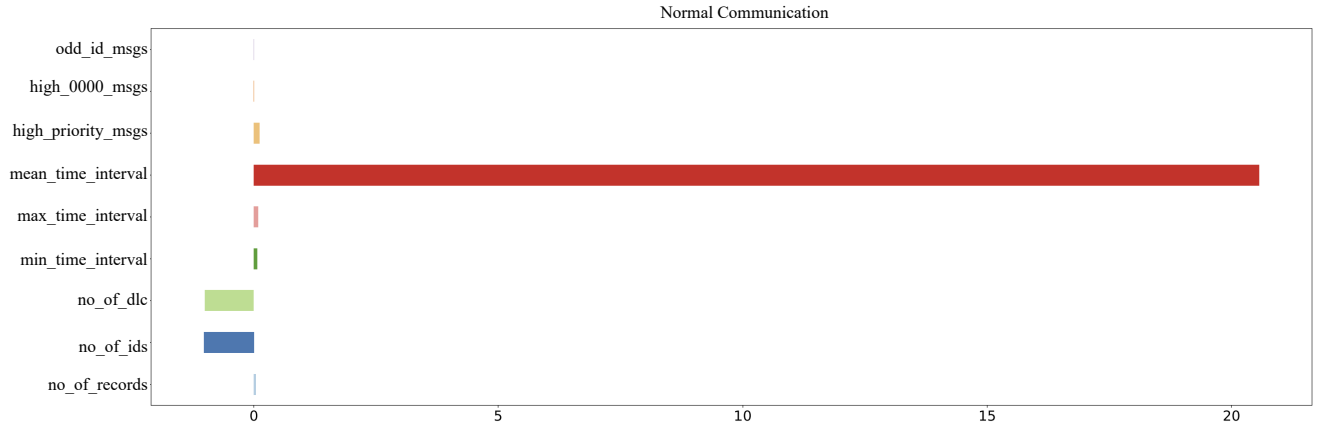


Fig. 7. Car Hacking Dataset: Outcomes generated using adversarial approach for normal CAN frame communication during attacks (DoS)

be different from baseline communication as attack behaviors result in pre and post-effects in the systems, resulting in deviations from the baseline behavior. These deviations seem to mainly result from higher mean time intervals between frames. Further, the number of unique IDs and DLC values seems low compared to the baseline. This matches domain experts' knowledge: attack communication results in a latency of CAN frames. These features need to be discussed with domain experts.

V. Conclusion

This work presented a ResNet Autoencoder based Explainable Anomaly Detection System (RX-ADS) for CAN bus communication data. RX-ADS consists of three components, 1) Window based feature extraction component to extract features on CAN bus communication, 2) Autoencoder Neural Network based anomaly detection system to detect anomalies in CAN communication, and 3) Adversarial machine learning based component for explaining the detected anomalies. The approach was tested on two benchmark CAN datasets (OTIDS and Car Hacking). RX-ADS showed similar or better results compared to state-of-the-art approaches on benchmark data and widely used anomaly detection algorithms. Thus, empirical results suggest that the proposed reconstruction error-based ADS together with the presented window-based feature extractor is a viable solution for detecting anomalous behaviors in EV CAN bus communication. Feature value distribution of abnormal communication data before and after applying the adversarial approach shows the generated adversarial samples have a much closer feature value distribution to the normal feature value distribution. I.e., Generated adversarial samples mimic normal feature value distributions. Thus, we can use the magnitude of modifications for generating the explanations for detected anomalies.

Based on the generated explanations, we have discovered the main characteristics distinguishing normal communication from attacks and between two tested attack types. We have verified that these characteristics match the opinions of domain experts. We identified that the characteristics that distinguish normal communication from attacks (DoS and Fuzzy) are a very low lost reply rate, a higher number of instant reply rates (request frame followed by a response frame), and a very low amount of high-priority/odd frames within a window. Characteristics that distinguish DoS from Fuzzy attacks are that DoS result in a higher number of high-priority messages (0000) whereas Fuzzy results in a higher number of random IDs which haven't encounter during normal behavior. I.e., the number of unique IDs is very high during fuzzy attacks compared to DoS.

Currently, the proposed approach is being deployed in a real-world setting: Idaho National Laboratory (INL) EV Charging System setup [20]. Further, the proposed approach will be extended to add physical features for providing more holistic abnormal behavior detection in EV infrastructure. A human study will be performed with domain experts to further validate the system performance in the charging system. We will perform further system improvements for addressing current system limitations.

Acknowledgements

This work was supported in part by the Department of Energy through the U.S. DOE Idaho Operations Office under Contract DE-AC07-05ID14517, and in part by the Commonwealth Cyber Initiative, an Investment in the Advancement of Cyber Research and Development, Innovation and Workforce Development (cyberinitiative.org).

References

[1] H. Zakaria, M. Hamid, E. M. Abdellatif, and A. Imane, "Recent advancements and developments for electric vehicle technology," in *2019 International Conference of Computer Science and Renewable Energies (ICCSRE)*, 2019, pp. 1–6. DOI: 10.1109/ICCSRE.2019.8807726.

[2] B. Singh and P. K. Dubey, "Distributed power generation planning for distribution networks using electric vehicles: Systematic attention to challenges and opportunities," *Journal of Energy Storage*, 2022.

[3] B. Singh, P. K. Dubey, and S. N. Singh, "Recent optimization techniques for coordinated control of electric vehicles in super smart power grids network: A state of the art," in *2022 IEEE 9th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, 2022, pp. 1–7. DOI: 10.1109/UPCON56432.2022.9986471.

[4] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383–398, 2019. DOI: 10.1109/TITS.2018.2815678.

[5] J. Guerrero-Ibáñez, S. Zeadally, and J. Contreras-Castillo, "Sensor technologies for intelligent transportation systems," *Sensors*, vol. 18, no. 4, 2018. DOI: 10.3390/s18041212.

[6] A. F. M. Boukerche and R. W. L. Coutinho, "Crowd management: The overlooked component of smart transportation systems," *IEEE Communications Magazine*, vol. 57, pp. 48–53, 2019.

[7] Y. Fraiji, L. Ben Azzouz, W. Trojet, and L. A. Saidane, "Cyber security issues of internet of electric vehicles," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, 2018, pp. 1–6. DOI: 10.1109/WCNC.2018.8377181.

[8] M. K. Hasan, A. A. Habib, S. Islam, M. Balfaqih, K. M. Alfawaz, and D. Singh, "Smart grid communication networks for electric vehicles empowering distributed energy generation: Constraints, challenges, and recommendations," *Energies*, vol. 16, no. 3, p. 1140, 2023.

[9] P. Franzese *et al.*, "Fast dc charging infrastructures for electric vehicles: Overview of technologies, standards, and challenges," *IEEE Transactions on Transportation Electrification*, 2023.

[10] E. Seo, H. M. Song, and H. K. Kim, "Gids: Gan based intrusion detection system for in-vehicle network," in *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, 2018, pp. 1–6. DOI: 10.1109/PST.2018.8514157.

[11] M. Aloqaily, S. Otoum, I. A. Ridhawi, and Y. Jararweh, "An intrusion detection system for connected vehicles in smart cities," *Ad Hoc Networks*, vol. 90, p. 101 842, 2019, Recent advances on security and privacy in Intelligent Transportation Systems. DOI: https://doi.org/10.1016/j.adhoc.2019.02.001.

[12] M. Basnet and M. Hasan Ali, "Deep learning-based intrusion detection system for electric vehicle charging station," in *2020 2nd International Conference on Smart Power Internet Energy Systems (SPIES)*, 2020, pp. 408–413. DOI: 10.1109/SPIES48661.2020.9243152.

[13] D. L. Marino, C. S. Wickramasinghe, and M. Manic, "An adversarial approach for explainable ai in intrusion detection systems," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 2018, pp. 3237–3243. DOI: 10.1109/IECON.2018.8591457.

[14] T. Jiang, J. L. Gradus, and A. J. Rosellini, "Supervised machine learning: A brief primer," *Behavior Therapy*, vol. 51, no. 5, pp. 675–687, 2020. DOI: https://doi.org/10.1016/j.beth.2020.05.002.

[15] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, Dec. 2019. DOI: 10.1186/s42400-019-0038-7.

[16] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," *Expert Systems with Applications*, vol. 41, no. 4, Part 2, pp. 1690–1700, 2014. DOI: https://doi.org/10.1016/j.eswa.2013.08.066.

[17] C. S. Wickramasinghe, D. L. Marino, and M. Manic, "Resnet autoencoders for unsupervised feature learning from high-dimensional data: Deep models resistant to performance degradation," *IEEE Access*, vol. 9, pp. 40 511–40 520, 2021. DOI: 10.1109/ACCESS.2021.3064819.

[18] C. S. Wickramasinghe, D. L. Marino, J. Grandio, and M. Manic, "Trustworthy ai development guidelines for human system interaction," in *2020 13th International Conference on Human System Interaction (HSI)*, 2020, pp. 130–136. DOI: 10.1109/HSI49210.2020.9142644.

[19] C. S. Wickramasinghe, K. Amarasinghe, D. L. Marino, C. Rieger, and M. Manic, "Explainable unsupervised machine learning for cyber-physical systems," *IEEE Access*, vol. 9, pp. 131 824–131 843, 2021. DOI: 10.1109/ACCESS.2021.3112397.

[20] H. S. Mavikumbure *et al.*, "Physical anomaly detection in ev charging stations: Physics-based vs resnet ae," in *In 2023 IEEE 32nd International Symposium on Industrial Electronics (ISIE)*, IEEE, 2023.

[21] H. Lee, S. H. Jeong, and H. K. Kim, "Otids: A novel intrusion detection system for in-vehicle network by using remote frame," in *2017 15th Annual Conference on Privacy, Security and Trust (PST)*, 2017, pp. 57–5709. DOI: 10.1109/PST.2017.00017.

[22] M. E. Verma, M. D. Iannacone, R. A. Bridges, S. C. Hollifield, B. Kay, and F. L. Combs, "ROAD: the real ORNL automotive dynamometer controller area network intrusion detection dataset (with a comprehensive CAN IDS dataset survey & guide)," *CoRR*, vol. abs/2012.14600, 2020. arXiv: 2012.14600.

[23] A. Tomlinson, J. Bryans, S. A. Shaikh, and H. K. Kalutarage, "Detection of automotive can cyber-attacks by identifying packet timing anomalies in time windows," in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2018, pp. 231–238. DOI: 10.1109/DSN-W.2018.00069.

[24] M. R. Moore, R. A. Bridges, F. L. Combs, M. S. Starr, and S. J. Prowell, "Modeling inter-signal arrival times for accurate detection of can bus signal injection attacks: A data-driven approach to in-vehicle intrusion detection," in *Proceedings of the 12th Annual Conference on Cyber and Information Security Research*, ser. CISRC '17, Oak Ridge, Tennessee, USA: Association for Computing Machinery, 2017. DOI: 10.1145/3064814.3064816.

[25] M. Hanselmann, T. Strauss, K. Dormann, and H. Ulmer, "Canet: An unsupervised intrusion detection system for high dimensional can bus data," *IEEE Access*, vol. 8, pp. 58 194–58 205, Mar. 2020. DOI: 10.1109/ACCESS.2020.2982544.

[26] W. Choi, K. Joo, H. J. Jo, M. C. Park, and D. H. Lee, "Voltageids: Low-level communication characteristics for automotive intrusion detection system," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 8, pp. 2114–2129, 2018. DOI: 10.1109/TIFS.2018.2812149.

[27] A. Taylor, S. Leblanc, and N. Japkowicz, "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)*, 2016, pp. 130–139. DOI: 10.1109/DSAA.2016.20.

[28] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PLoS ONE*, vol. 11, 2016.

[29] M. Delwar Hossain, H. Inoue, H. Ochiai, D. Fall, and Y. Kadobayashi, "An effective in-vehicle can bus intrusion detection system using cnn deep learning approach," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–6. DOI: 10.1109/GLOBECOM42002.2020.9322395.

[30] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *CoRR*, vol. abs/1611.01236, 2016. arXiv: 1611.01236.

[31] R. S. Siva Kumar *et al.*, "Adversarial machine learning-industry perspectives," in *2020 IEEE Security and Privacy Workshops (SPW)*, 2020, pp. 69–75. DOI: 10.1109/SPW50608.2020.00028.

[32] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," Apr. 2017, pp. 506–519. DOI: 10.1145/3052973.3053009.

[33] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, 2018. DOI: https://doi.org/10.1016/j.patcog.2018.07.023.

[34] K. Amarasinghe, C. Wickramasinghe, D. Marino, C. Rieger, and M. Manicl, "Framework for data driven health monitoring of cyber-physical systems," in *2018 Resilience Week (RWS)*, 2018, pp. 25–30. DOI: 10.1109/RWEEK.2018.8473535.

[35] A. Derhab, M. Belaoued, I. Mohiuddin, F. Kurniawan, and M. K. Khan, "Histogram-based intrusion detection and filtering framework for secure and safe in-vehicle networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2021. DOI: 10.1109/TITS.2021.3088998.

[36] I. Berger, R. Rieke, M. Kolomeets, A. Chechulin, and I. Kotenko, "Comparative study of machine learning methods for in-vehicle intrusion detection," in *CyberICPS/SECPRE@ESORICS*, 2018.

Chathurika S. Wickramasinghe received her B.Sc. degree in computer science from the University of Peradeniya, Sri Lanka, in 2016; And her doctoral degree in computer science at Virginia Commonwealth University, Richmond, USA. Currently, she is working as a Sr Assoc, Data Science at Capital One, USA. Her research interests include machine learning, unsupervised learning, explainable AI, generalization, and visual data mining.

Daniel L. Marino received his B.Eng. in automation engineering from La Salle University, Colombia, in 2015; And his doctoral degree at Virginia Commonwealth University, Richmond, USA. Currently, he is working as an Applied Scientist in Amazon, USA. His research interests include stochastic modeling, deep learning, and optimal control.

Harindra S. Mavikumbure (Student Member, IEEE) received his B.Sc. degree in computer science from the University of Peradeniya, Sri Lanka, in 2018. He is currently working as a research assistant while reading for his doctoral degree in computer science at Virginia Commonwealth University, Richmond. His research interests include Anomaly Detection, Machine Learning, Deep Learning, Unsupervised Learning.

Victor Cobilean received his B.Sc. degree in mechatronics from the Technical University of Cluj-Napoca, Romania, in 2020. He recieved his M. Sc. in Mechatronic Systems Engineering from Technical University of Cluj-Napoca, Romania, in 2022.He is currently reading for his doctoral degree in computer science at Virginia Commonwealth University, Richmond. His research interests include Anomaly Detection, Informed Machine Learning, Unsupervised Learning.

Timothy Pennington is a Senior Research Engineer and Group Lead for the Vehicle Grid Integration (VGI) Research Group within the Energy Storage & Advanced Transportation Department of the Idaho National Laboratory (INL). Tim's research and team investigates the charging of Electric Vehicles (EV), and specifically their interactions with the electric grid. Current projects and capabilities focus on three areas, including Wireless Charging – both stationary and in-motion charging, Cybersecurity of EV Charging Infrastructure, and Charging Energy Forecast and Management. Prior to joining INL, Tim was a civilian engineer with the Navy and led Department of Defense research and technology demonstration projects. Tim holds a BSc from the Massachusetts Institute of Technology and a MSc from the University of Southampton (UK).

Benny Varghese is a research engineer at the Idaho National Laboratory working on EV Charging Infrastructure. He is currently a working group chair at the National ChargeX Consortium, funded by the Joint Office of Energy and Transportation to improve EV charging experience. In his current role, he is working to improve the reliability of public DC fast charging infrastructure and remove roadblocks to EV adoption. His areas of expertise include testing, design and development of EV charging infrastructure. Benny has a Ph.D. from Utah State University, where he developed static and dynamic wireless charging solutions for electric vehicles.

Craig Rieger (Senior Member, IEEE) received the B.S. and M.S. degrees in chemical engineering from Montana State University, Bozeman, MT, USA, in 1983 and 1985, respectively, and the Ph.D. degree in engineering and applied science from Idaho State University, Pocatello, ID, USA, in 2008. He is currently the Chief Control Systems Research Engineer and a Directorate Fellow with Idaho National Laboratory (INL), Idaho Falls, ID, USA, pioneering interdisciplinary research in the area of next-generation resilient control systems. In addition, he has organized and chaired the 11 Institute of Electrical and Electronics Engineers technically cosponsored symposia and one NSF Workshop. He has also been a Supervisor and a Technical Lead for control systems engineering groups for several INL nuclear facilities and various control system architectures.

Milos Manic (SM'06-M'04-StM'96) is a Professor with the Computer Science Department and Director of VCU Cybersecurity Center at Virginia Commonwealth University. He completed over 50 research grants in AIML in cyber and energy and intelligent controls. He authored over 200 refereed articles, has given over 50 invited talks around the world, authored over 200 refereed articles in international journals, books, and conferences, holds several U.S. patents, and has won the 2018 R&D 100 Award for Autonomic Intelligent Cyber Sensor (AICS), one of top 100 science and technology worldwide innovations in 2018. He is an inductee of the US National Academy of Inventors (senior class of 2023, member class of 2019), and a Fellow of the Commonwealth Cyber Initiative (specialty in AI & Cybersecurity). He holds Joint Appointment with the Idaho National Laboratory. He is an IEEE IES President-Elect, IEEE Fellow (for contributions to machine learning-based cybersecurity in critical infrastructures), Senior Inductee of the US National Academy of Inventors (NAI), and recipient of the IEEE IES 2019 Anthony J. Hornfeck Service Award, 2012 J. David Irwin Early Career Award, 2017 IEM Best Paper Award, associate editor of Transactions on Industrial Informatics, Open Journal of Industrial Electronics Society, and Senior Life AdCom member.