Computers & Security 151 (2025) 104322

Contents lists available at ScienceDirect

Computers & Security

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

Full length article

Enhancing can security with ML-based IDS: Strategies and efficacies against adversarial attacks

Ying-Dar Lin [a,*], Wei-Hsiang Chan [a], Yuan-Cheng Lai [b], Chia-Mu Yu [c], Yu-Sung Wu [a], Wei-Bin Lee [d]

[a] Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, 300, Taiwan
[b] Department of Information Management, National Taiwan University of Science and Technology, Taipei, 10607, Taiwan
[c] Department of Electronics and Electrical Engineering, National Yang Ming Chiao Tung University, Hsinchu, 300, Taiwan
[d] Hon Hai Research Institute, Taipei, Taiwan

ARTICLE INFO

Keywords:
Adversarial attack
Machine learning
Intrusion detection
Distance-based optimization
Electronic vehicle

ABSTRACT

Control Area Networks (CAN) face serious security threats recently due to their inherent vulnerabilities and the increasing sophistication of cyberattacks targeting automotive and industrial systems. This paper focuses on enhancing the security of CAN, which currently lack adequate defense mechanisms. We propose integrating Machine Learning-based Intrusion Detection Systems (ML-based IDS) into the network to address this vulnerability. However, ML systems are susceptible to adversarial attacks, leading to misclassification of data. We introduce three defense combination methods to mitigate this risk: adversarial training, ensemble learning, and distance-based optimization. Additionally, we employ a simulated annealing algorithm in distance-based optimization to optimize the distance moved in feature space, aiming to minimize intra-class distance and maximize the inter-class distance. Our results show that the ZOO attack is the most potent adversarial attack, significantly impacting model performance. In terms of model, the basic models achieve an F1 score of 0.99, with CNN being the most robust against adversarial attacks. Under known adversarial attacks, the average F1 score decreases to 0.56. Adversarial training with triplet loss does not perform well, achieving only 0.64, while our defense method attains the highest F1 score of 0.97. For unknown adversarial attacks, the F1 score drops to 0.24, with adversarial training with triplet loss scoring 0.47. Our defense method still achieves the highest score of 0.61. These results demonstrate our method's excellent performance against known and unknown adversarial attacks.

1. Introduction

Electric vehicles (EVs) are growing in popularity and rely on multiple electronic control units (ECUs) for critical functions like braking and steering. These ECUs communicate with each other via the Control Area Network (CAN) (Natale et al., 2012), which lacks strong security features like encryption or authentication (Miller and Valasek, 2013). This weakness enables hackers to access the CAN and send unauthorized commands, potentially manipulating vehicle behavior and compromising safety and operation.

Given the vulnerabilities in the CAN, implementing an Intrusion Detection System (IDS) is essential for security. Although anomaly-based and signature-based IDS have limitations in recognizing anomalies and known threats, a machine learning (ML)-based IDS is more effective. It efficiently detects malicious CAN messages and helps identify new types of attacks. However, it is crucial to note that ML-based IDS can still be susceptible to adversarial attacks.

Adversarial attacks (Wang, 2018; Martins et al., 2020), particularly evasion attacks that alter test data to deceive ML models, pose a major threat to ML systems, including our ML-based CAN IDS. These attacks can trick the IDS, leading to compromised ECUs and vulnerabilities in the CAN network. To counter this threat, implementing strong defenses is crucial to protect against such attacks and preserve the integrity of the CAN network.

To counter adversarial attacks, Adversarial Training (AT) (Miyato et al., 2015) and Ensemble Learning (EL) are commonly used. AT retrains models with adversarial data to improve resistance to known attacks, while EL (Strauss et al., 2017) uses multiple models to enhance decision-making through diverse data interpretations. However, these

* Corresponding author.
E-mail addresses: ydlin@cs.nctu.edu.tw (Y. Lin), ivan89031580.cs11@nycu.edu.tw (W. Chan), laiyc@cs.ntust.edu.tw (Y. Lai), chiamuyu@nycu.edu.tw (C. Yu), ysw@nycu.edu.tw (Y. Wu), wei-bin.lee@foxconn.com (W. Lee).

https://doi.org/10.1016/j.cose.2025.104322
Received 12 August 2024; Received in revised form 14 December 2024; Accepted 6 January 2025
Available online 23 January 2025

methods might not fully protect against new, unknown attacks, making the development of further defenses essential to effectively address emerging threats.

Another defensive approach is Distance-based Optimization (DO), aiming to keep feature vectors sufficiently distant from each class to ensure accurate classification despite noise. Determining the optimal distance and direction is challenging, often relying on heuristics, which may not yield the best solution. To overcome this, Simulated Annealing (SA) (Bertsimas and Tsitsiklis, 1993), a heuristic and global optimization algorithm, is employed to identify optimal parameters. By integrating adversarial training, ensemble learning, and distance-based optimization, the CAN IDS system can effectively counter both known and unknown adversarial threats, maintaining robust network security.

Adversarial defense research typically centers on network IDS (Anthi et al., 2021; Demontis et al., 2019; Lin et al., 2022) and image recognition (Chien and Chen, 2024; Mao et al., 2019; Li et al., 2019; Deng and Mu, 2024; Mustafa et al., 2020; Seo et al., 2023). Our focus, however, is on EV emulation, which faces significant challenges from adversarial attacks. Common defenses like adversarial training (Miyato et al., 2015) and ensemble learning (Strauss et al., 2017) might not thwart unknown adversarial attacks. Recently, distance-based methods (Wen et al., 2016) have been explored, though they are still emerging and under-researched. Electric vehicles (EVs) are growing in popularity and rely on multiple electronic control units (ECUs) for critical functions like braking and steering. These ECUs communicate with each other via the Control Area Network (CAN) (Ashraf and Ahmed, 2020), which lacks strong security features like encryption or authentication (Miller and Valasek, 2013). This weakness enables hackers to access the CAN and send unauthorized commands, potentially manipulating vehicle behavior and compromising safety and operation.

Given the vulnerabilities in the CAN, implementing an Intrusion Detection System (IDS) is essential for security. Although anomaly-based and signature-based IDS have limitations in recognizing anomalies and known threats, a machine learning (ML)-based IDS is more effective. It efficiently detects malicious CAN messages and helps identify new types of attacks. However, it is crucial to note that ML-based IDS can still be susceptible to adversarial attacks.

Adversarial attacks (Wang, 2018; Martins et al., 2020), particularly evasion attacks that alter test data to deceive ML models, pose a major threat to ML systems, including our ML-based CAN IDS. These attacks can trick the IDS, leading to compromised ECUs and vulnerabilities in the CAN network. To counter this threat, implementing strong defenses is crucial to protect against such attacks and preserve the integrity of the CAN network.

To counter adversarial attacks, Adversarial Training (AT) (Miyato et al., 2015) and Ensemble Learning (EL) are commonly used. AT retrains models with adversarial data to improve resistance to known attacks, while EL (Strauss et al., 2017) uses multiple models to enhance decision-making through diverse data interpretations. However, these methods might not fully protect against new, unknown attacks, making the development of further defenses essential to effectively address emerging threats.

Another defensive approach is Distance-based Optimization (DO), aiming to keep feature vectors sufficiently distant from each class to ensure accurate classification despite noise. Determining the optimal distance and direction is challenging, often relying on heuristics, which may not yield the best solution. To overcome this, Simulated Annealing (SA) (Bertsimas and Tsitsiklis, 1993), a heuristic and global optimization algorithm, is employed to identify optimal parameters. By integrating adversarial training, ensemble learning, and distance-based optimization, the CAN IDS system can effectively counter both known and unknown adversarial threats, maintaining robust network security.

Adversarial defense research typically centers on network IDS (Anthi et al., 2021; Demontis et al., 2019; Lin et al., 2022) and image recognition (Chien and Chen, 2024; Mao et al., 2019; Li et al., 2019; Deng and Mu, 2024; Mustafa et al., 2020; Seo et al., 2023). Our focus, however, is on EV emulation, which faces significant challenges from adversarial attacks. Common defenses like adversarial training (Miyato et al., 2015) and ensemble learning (Strauss et al., 2017) might not thwart unknown adversarial attacks. Recently, distance-based methods (Wen et al., 2016) have been explored, though they are still emerging and under-researched. Our goal is to combine these methods to form a comprehensive defense against both known and unknown adversarial attacks. Our goal is to combine these methods to form a comprehensive defense against both known and unknown adversarial attacks. This paper addresses two main concerns. Firstly, it examines distance optimization, noting that while heuristic methods are useful, they may not always provide the best solution. The objective is to determine the optimal distance and direction for movement. Secondly, the paper aims to identify the most effective ML-based model for CAN IDS. With three defensive approaches available, it seeks to assess and determine the best single or combined strategy. The goal is to comprehensively evaluate all defense combinations to identify the optimal ML-based CAN IDS solution.

Several factors impede the direct adaptation of state-of-the-art (SOTA) image defenses to the CAN bus domain. First, in contrast to the unstructured nature of image data, where individual pixels may lack standalone significance, CAN data is structured, with each element, such as timestamp, identifier (ID), and data fields, holding specific meanings. Second, though CAN messages and images share a data range of 0 to 255, their data types differ significantly; CAN fields are typically integers, while image data often uses floating-point values. Furthermore, CAN IDs can include up to 29 bits, and timestamps are represented with floating-point values, unlike image data that does not mix floating points and integers within the same dataset. These fundamental differences suggest that directly applying image-based defense strategies to the CAN bus may not be effective.

There are two main contributions:

- A paper has implemented distance-based optimization in image recognition (Seo et al., 2023), but this approach has not been applied in an EV environment. We not only adapt it to the EV setting but also develop our own distance-based optimization algorithm.

- Many researchers employ adversarial defense techniques like adversarial training and ensemble learning on CAN, but these methods primarily protect against known adversarial attacks. We extend beyond these strategies by incorporating distance-based optimization to defend against unknown adversarial threats. Additionally, we validate our approach through EV emulation testing.

This paper is organized as follows: Section 2 reviews background and related work; Section 3 outlines notations and problem statements; Section 4 discusses solution approaches; Section 5 details our solution implementation on EVπ; Section 6 evaluates experimental results; Section 7 concludes and suggests future directions.

2. Background and related works

2.1. Control Area Network (CAN) & CAN threat

A standard CAN data frame has several key components: the Start of Frame (SOF) signals the start of transmission; the Arbitration field includes the Identifier (ID) and Remote Transmission Request (RTR) for message prioritization and type distinction; the control section features the Identifier Extension (IDE), a reserved bit (r0), and the Data Length Code (DLC) that specifies payload size. The Data field carries actual information. Limited to an 8-byte payload, CAN frames lack encryption and authentication, exposing them to security risks like network disruption or unauthorized system access, necessitating robust risk mitigation measures.

Attackers compromise vehicles via physical OBD-II ports or remote access, connecting through OBD-II or the in-car entertainment system. The unencrypted and unauthenticated CAN network allows direct injection of malicious data, posing threats like unauthorized information access, vehicle control loss, and passenger danger. Tactics include DDoS or spoofing attacks, potentially disabling brakes, controlling acceleration, or altering CAN bus messages.

2.2. Adversarial attack

Adversarial attacks originated in the image domain (Goodfellow et al., 2014) and expanded to networks, but are less common in the CAN bus domain. We treat CAN data frames as features, inducing perturbations by altering data points, causing potential misclassifications by machine learning algorithms through noise addition.

Adversarial attacks vary in type and method, including gradient-based attacks like Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014), Projected Gradient Descent (PGD) (Madry et al., 2017), Momentum Iterative Method (MIM) (Dong et al., 2018), and Jacobian Saliency Map (JSAM) (Papernot et al., 2016), which modify inputs using model gradients. Optimization-based attacks, such as DeepFool (Moosavi-Dezfooli et al., 2016) and ZOO (Chen et al., 2017), minimally alter data to cause misclassifications.

Attacks differ in iteration needs: single-step attacks like FGSM require one iteration, while multi-step attacks like PGD and JSMA use several, increasing computational demands. Moreover, attack portability enables adversarial examples from one model to compromise others, enhancing attack feasibility without direct model access.

2.3. Adversarial defense

The necessity for adversarial defense stems from the susceptibility of machine learning models to adversarial attacks, threatening their performance and reliability. In our ML-based CAN IDS, robust defense against these attacks is crucial for maintaining safety and security.

- **Adversarial Training**: Adversarial training (Miyato et al., 2015) uses adversarial examples generated by attacks for training, enhancing model robustness by providing a different data perspective.
- **Ensemble Learning**: Ensemble learning (Strauss et al., 2017) combines multiple models to utilize their diverse perspectives, evaluating their collective performance with metrics like the kappa statistic for model agreement. This method mitigates "double error" risks — when all models err identically — by leveraging the combined insights and diversity of the models for a thorough evaluation of test data.
- **Distance-based Optimization**: Distance-based optimization enhances model robustness by minimizing intra-class distance and maximizing inter-class distance through two phases: the shrink phase and the push-back (PB) phase. The shrink phase (Seo et al., 2023) reduces intra-class distance by aligning class feature vectors towards their centers, while the PB phase increases inter-class distance by separating classes further. However, the heuristic methods for distance calculation might not be optimal. Optimal intra-class compactness and inter-class separability are crucial for robust adversarial defense, underscoring the need for advanced optimization techniques.

2.4. Simulated annealing

Heuristic-based distance optimization hinders finding the optimal defense distance. To overcome this, we apply Simulated Annealing (SA) (Bertsimas and Tsitsiklis, 1993), a heuristic and probabilistic optimization algorithm inspired by metallurgical annealing. SA solves global optimization problems by decreasing temperature until convergence, escaping local optima using temperature-dependent probability to reach global solutions.

In this study, we did not select alternative optimization techniques because classical optimization methods would likely yield similar improvements. The objective was to compare heuristic or non-optimized distance-based methods with Distance-based Optimization (DO), rather than to evaluate different optimization algorithms. For this purpose, we chose simulated annealing, a widely used and representative algorithm.

2.5. Related work

Table 1 highlights previous research on adversarial attacks and defenses. In adversarial defense, numerous papers focus on adversarial training (Demontis et al., 2019; Lin et al., 2022; Chien and Chen, 2024; Mao et al., 2019; Li et al., 2019; Deng and Mu, 2024; Mustafa et al., 2020; Wang et al., 2023). Some studies employ ensemble learning (Lin et al., 2022; Wang et al., 2023). In the realm of distance-based methods, some shrink techniques are used (Mustafa et al., 2020; Seo et al., 2023). However, our approach differs in that it implements shrink techniques with optimization methods to find the best distance to move. In adversarial attack, there are common adversarial attack techniques such as FGSM, JSMA, PGD, and ZOO are also discussed.

In Demontis et al. (2019), Lin et al. (2022), researchers utilize simple models such as decision trees (DT), random forests (RF), support vector machines (SVM), and deep neural networks (DNN) in the network domain. Additionally, studies like (Chien and Chen, 2024; Mao et al., 2019; Li et al., 2019; Deng and Mu, 2024; Mustafa et al., 2020; Seo et al., 2023) incorporate Adversarial Contrastive Learning (ACL), Robust Contrastive Learning (RoCL), Adversarial Contrastive Learning framework (ADVCL), Convolutional Neural Networks (CNN) and DNN models, primarily focusing on image recognition tasks. In Wang et al. (2023), various machine learning models, including DT and SVM, are employed in EV simulations. Transferability analysis is performed exclusively in Lin et al. (2022), Wang et al. (2023).

‘‘‘

Our research uses neural network (NN) models for defense, focusing on distance-based optimization tailored for NN models. We evaluate all adversarial defense techniques in the corresponding work table to determine the most effective approach. Notably, our research focuses on EVs, and we conduct emulation experiments. This choice allows us to more accurately simulate real-world conditions, providing valuable insights into the effectiveness of our defense strategies in the EV domain. In Wang et al. (2023), the work is the most similar to ours, but there are key differences. While we discuss adversarial attack and defense in EV emulation, Wang et al. (2023) focuses on EV simulation. The difference in environment is significant, as our approach is closer to real-world conditions. Additionally, our defense approaches differ; although both studies use adversarial training and ensemble methods, our work uniquely incorporates distance-based optimization.

3. Problem statements

This section covers our discussion of the problem statement in two subsections: the notation table and the problem description. We start by explaining the notations. Next, we describe the main problem of the work.

3.1. Notations

The notation table is divided into three categories: Dataset, Machine Learning, and Distance-based. The Dataset category includes the clean and adversarial datasets, with the expanded dataset being a mix of both. The Machine Learning category encompasses various models with different adversarial defense methods, such as adversarial training and

Table 1
Survey on Adversarial Defense.

Paper	Adversarial training	Ensemble learning	Distanced		Attack techniques	Models	Transferability analysis	Environment
			shrink	PB				
Demontis et al. (2019)	V	–	–	–	JSMA	Random Forest, J48	–	Network
Lin et al. (2022)	V	V	–	–	Decision Tree Attack, JSMA, ZOO Attack, C&W, PGD, FGSM	Decision Tree, LR, XGBoost, SVM, DNN	V	Network
Chien and Chen (2024)	V	–	–	–	PGD	ACL, RoCL, ADVCL	–	Image recognition
Mao et al. (2019)	V	–	–	–	FGSM and C&W, LL	DNN	–	Image recognition
Li et al. (2019)	V	–	–	–	FGSM, BIM, PGD, C&W, MIM	DNN	–	Image recognition
Deng and Mu (2024)	V	V	–	–	AutoAttack, PGD, C&W, SignHunter	DNN	–	Image recognition
Mustafa et al. (2020)	V	–	V	–	FGSM, BIM, MIM, PGD, C&W	CNN	–	Image recognition
Seo et al. (2023)	–	–	V	–	FGSM, BIM, MIM, PGD	CNN	–	Image recognition
Wang et al. (2023)	V	V	–	–	ZOO Attack, JSMA, PGD, FGSM, C&W, BIM, DTA	Decision Tree, SVM, LSTM, XGB, DNN, CNN, LR, KNN	V	EV Simulation
Ours	V	V	V	V	JSMA, Deepfool, MIM, FGSM, PGD, ZOO Attack	DNN, CNN, LSTM	V	EV Emulation

distance-based optimization, as well as ensemble models. The Distance-based category includes the distance algorithm, feature vectors, center vector, and movement distance. Table 2 lists all the notations in these categories.

‘‘‘

Dataset: A prepared dataset D will be used to train the ML-based CAN IDS. This dataset contains the input data x_i and its corresponding label y_i. The dataset is divided into the training dataset D^{Train} and the testing dataset D^{Test}. Additionally, adversarial attacks can attack the dataset, resulting in the adversarial dataset D^{+} contains input data x_i^{+} and its corresponding label y_i. The adversarial dataset includes a well-known adversarial dataset D^{kn+} for adversarial training and an unknown adversarial dataset D^{un+} for testing the model's robustness. During adversarial training, we need an expanded dataset D^{Ex}, which combines the original dataset D and the well-known adversarial dataset D^{kn+}. This expanded dataset is split into the expanded training dataset $D^{ExTrain}$ and the expanded testing dataset $D^{ExTrain}$

Machine Learning: The best model M^{*} is trained on dataset D and evaluated using the F1 score. The best adversarial training model M^{+*} follows the same evaluation but is trained with the expanded dataset $D^{ExTrain}$. The best distance-based optimization model M^{D*} uses the same dataset and evaluation as M^{*}, but incorporates distance-based optimization. The best distance-based optimization with adversarial training model M^{+D*} is trained with $D^{ExTrain}$ and uses distance-based optimization. Finally, the best ensemble models — E^{*}, E^{+*}, E^{D*}, and E^{+D*} — are formed by enhancing these previous models with ensemble learning.

Distance-based: Distance-based optimization consists of a shrink phase and a push-back phase, each with its respective algorithms: the shrink algorithm $S(x)$ and the push-back algorithm $PB(x)$. The distance difference function calculates the total distance moved from the original feature vector v_i^j to the expected feature vector $v_i^{j'}$. In the shrink phase, we use the center vector C and feature vector v_i^j determines the direction $\overrightarrow{s_i^{j'}}$ and distance d_s for moving the feature vector. Similarly, in the push-back phase, center vector C and feature vector v_i^j are also used to calculate the direction $\overrightarrow{c^{j'}}$ and distance d_{PB}. Simulated annealing (SA) is then employed to select the best distance d_{best} to move.

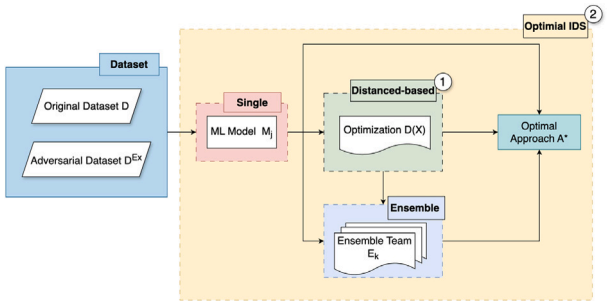

Fig. 1. Problem Overview. The figure depicts the process of evaluating single ML models and ensemble methods using both original and adversarial datasets. It highlights the integration of distance-based optimization and ensemble strategies to identify the optimal IDS approach A^{*}.

3.2. Problem statements

3.2.1. Problem overview

Fig. 1 illustrates our research framework. We use clean and adversarial datasets for adversarial training of individual models. Connections are made with these models to explore three adversarial defense approaches and identify the optimal strategy. Two main problems arise: distance optimization, which involves finding the best direction and distance to separate feature vectors, and achieving an optimal Intrusion Detection System (IDS) by determining the most effective machine learning integration approaches.

3.2.2. Problem 1 - Optimization for distance

Distance-based optimization consists of Shrink and PB, aiming to find the optimal distance and PB direction for movement to maximize the total distance.

Given the shrink function $S(x)$, PB function $PB(x)$, model M, and feature vectors of each model V as input, our goal is to decide the distance d_s, d_{PB}, and direction $\vec{c}$ as output. This is equivalent to

Table 2
Notations.

Dataset		
Dataset	D	$D = \{(x_i, y_i), i = 1, \ldots, n\}$; $D = D^{Test} \cup D^{Train}$; $Test \cup Train = \{1, \ldots, n\}$
Training Dataset	D^{Train}	$D^{Train} = \{(x_i, y_i), i \in Test\}$
Testing Dataset	D^{Test}	$D^{Test} = \{(x_i, y_i), i \in Train\}$
Adversarial Attack Data	x_i^+	$x_i^+ \in R^n$
Well-known Adversarial Attack Dataset	D^{kn+}	$D^{kn+} = \{(x_i^+, y_i), i = 1, \ldots, n\}$
Unknown Adversarial Attack Dataset	D^{un+}	$D^{un+} = \{(x_i^+, y_i), i = 1, \ldots, n\}$
Adversarial Attack Dataset	D^+	$D^+ = D^{kn+} \cup D^{un+}$
Well-known Adversarial Dataset for Testing	$D^{knTest+}$	$D^{knTest+} = \{(x_i^+, y_i), i \in Test\}$ $D^{knTest+} \in D^{kn+}$
Unknown Adversarial Dataset for Testing	$D^{unTest+}$	$D^{unTest+} = \{(x_i^+, y_i)\ i \in Test\}$, $D^{unTest+} \in D^{un+}$
Expanded Dataset with Adversarial Samples	D^{Ex}	$D^{Ex} = D \cup D^{kn+}$
Expanded Dataset for Training	$D^{ExTrain}$	$D^{ExTrain} = \{(x_i, y_i) \cup (x_i^+, y_i), i \in Train\}$
Expanded Dataset for Testing	D^{ExTest}	$D^{ExTest} = \{(x_i, y_i) \cup (x_i^+, y_i), i \in Test\}$
Machine Learning		
All Single Models	M	$M = M_j \cup M_j^+$
ML Model	M_j	Model training with Training Dataset D^{Train}
ML Model with Adversarial Training	M_j^+	Model training with Expanded Training Dataset $D^{ExTrain}$
Distanced-based ML Model	M_j^D	Model doing Distanced-based optimization
Best Single ML Model	M^*	Best F1 score Model
Best ML Model with Adversarial Training	M^{+*}	Best F1 score Adversarial Trained Model
Best ML Model with Distance-based Optimization	M^{D*}	Best F1 score Distance-based Optimization Model
Ensemble Team	E_k	E_k is consist of k M_j, $k = 2i + 1$, $i \in Z^+$
Best Ensemble Team	E^*	Best F1 score Ensemble Team
Best Adversarial Training Ensemble Team	E^{+*}	Best F1 score Adversarial Trained Ensemble Team
Best Distance-based Optimization Ensemble Team	E^{D*}	Best F1 score Distance-based Optimization Ensemble Team
Best Distance-based Optimization with Adversarial Training Ensemble Team	E^{+D*}	Best F1 score Distance-based Optimization with Adversarial Trained Ensemble Team
Best Approach	A^*	The smallest F1 scores degradation with testing dataset $D^{knTest+}$ and $D^{unTest+}$
Distance-based Optimization		
Shrink algorithm	$S(x)$	$x \in M$
PB algorithm	$PB(x)$	$x \in M$
Distance Difference function	$Dis(x)$	$x \in \{ d_s, d_{PB} \}$
Each feature vector	v_i^j	Feature vector with j class i elements.
Expected each feature vector	$v_i^{j'}$	Expected movement feature vector with j class i elements.
Feature vector	V	Feature vector of ML model, $V = \{(v_1^1, v_2^1, \ldots), (v_1^2, v_2^2, \ldots), \ldots\}$
Movement direction in simulated annealing	$\overline{c}$	Shrink and PB phase direction of vector of ML model.
Shrink phase movement direction in j class and i element	$\overrightarrow{s_i^{j'}}$	Shrink direction of vector of ML model in j class and i element.
Central vector	C	Central vector of ML model.
Central vector of j class	c^j	Central vector of ML model in j class.
PB phase movement direction of j class	$\overrightarrow{c^{j'}}$	PB direction of j class vector of ML model.
Distance of feature vector	d	Summation of every class feature to other class central vectors.
Shrink distance of feature vector	d_s	The distance we move in shrink phase calculated by SA.
PB distance of feature vector	d_{PB}	The distance we move in PB phase calculated by SA.
New distance to move	d_i	The new distance in each iteration in SA.
Best distance to move	d_{Best}	The best movement distance that is calculated by SA.
Threshold distance	δ^D	The average distance between each class

maximizing the distance

$$d = \frac{1}{n}\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{m}\sqrt{(v_i^j - c^k)^2}, i \neq k, \tag{1}$$

subject to the constraint that d_{PB} is less than or equal to constraint distance, denoted as Δ^D, which k is the center class number, j is the elements in each class, i is each class number. The objective function uses Euclidean distance to calculate the average distance from each class center to other class points, with constraints set to prevent infinite distance.

3.2.3. Problem 2 - Optimal ML-based model for CAN IDS

In this problem, we are presented with several defense approaches combined by distance-based optimization, adversarial training, and ensemble learning: the best ML model, denoted as M^*, the best adversarial training ML model, denoted as M^{+*}, the best Distance-based optimization ML model, denoted as M^{D*}, and the best Distance-based optimization with adversarial training ML model, denoted as M^{+D*}. Also, we have the best ensemble team, denoted as E^*, the best adversarial training ensemble team, denoted as E^{+*}, the best distance-based optimization ensemble team, denoted as E^{D*}, and the distance-based optimization with adversarial training ensemble team, denoted as E^{+D*}.

Our objective is to identify the optimal approach to serve as the IDS. Given the best models with adversarial defense combination methods listed above and additional datasets including the Expanded Testing Dataset $D^{ExTest+}$, and the Unknown Adversarial Attacked Testing Dataset $D^{unTest+}$ as input. Our task is to determine the best defense approach, denoted as A^*. This approach should minimize the degradation of the F1 score when testing with $D^{knTest+}$ and $D^{unTest+}$.

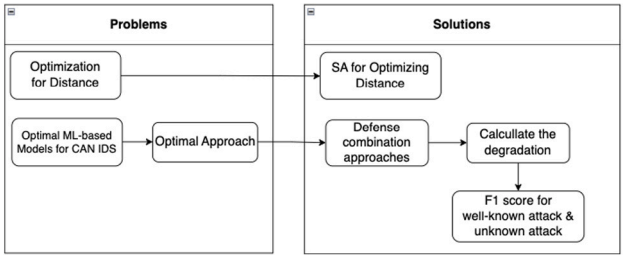

Fig. 2. Solution Overview. This figure outlines the proposed solutions to address the identified problems, including distance optimization using Simulated Annealing (SA) and evaluating defense combinations. The effectiveness of the approaches is measured by calculating the F1 score for both well-known and unknown attacks.

4. Solution approaches

4.1. Solution overview

In Fig. 2, we propose two solutions: (1) using SA to optimize distance and direction for feature vector movement, and (2) exploring an optimal ML-based CAN IDS solution using adversarial training, ensemble learning, and distance-based optimization. By combining these methods and evaluating their performance with adversarial test datasets, we aim to identify the best ML-based IDS solution for CAN.

4.2. Solution 1 - SA for optimizing distance

Fig. 3 shows the SA approach for distance optimization, combining Shrink (minimizing intra-distance) and PB (maximizing inter-distance) phases. SA generates new solutions with random variations, escaping local optima by accepting new solutions based on a probability criterion.

In the shrink phase, SA randomly generates d_i based on an interval number. We calculate the direction $\vec{c}$ to move, the difference $Dis(d_i)$ of features with a moving distance d_i in the direction $\vec{c}$ and the same class center features c^j, and the difference ΔD between $Dis(d_i)$ and the best solution distance $Dis(d_{Best})$, where $Dis(x)$ is the objective function of problem 1. If $Dis(d_i)$ is less than the current best difference distance $Dis(d_{Best})$, we accept $Dis(d_i)$ as the current best solution, or with a probability of $\frac{1}{e^{\frac{\Delta D}{t}}}$.

In the PB phase, we calculate the direction $\vec{c}$ to move, the difference $Dis(d_i)$ of each class's features v_i with a moving distance d_i in the direction $\vec{c}$ and other class center features c^j where $i \neq j$, and the difference ΔD between $Dis(d_i)$ and the best solution distance $Dis(d_{Best})$. If $Dis(d_i)$ is greater than the current best distance $Dis(d_{Best})$ and d_i is less than the constraint d_{thres}, we accept $Dis(d_i)$ as the current best solution, or with a probability of $\frac{1}{e^{\frac{\Delta D}{t}}}$.

Simulated temperature t and cooling parameter α are used to determine the temperature of each iteration. Worse solutions are more likely to be accepted at higher temperatures in probability $\frac{1}{e^{\frac{\Delta D}{t}}}$. Conversely, at lower temperatures, only reasonable solutions are accepted. The iteration terminates when the temperature t reaches the termination temperature t_{min}.

4.3. Solution 2 - Optimal ML-based CAN IDS model

4.3.1. General framework

Figs. 4 shows that our solution framework integrates three defense methodologies, resulting in eight approaches: without any adversarial defense, individual defenses (DO, AT, EL), combinations of two defenses (DO+AT, AT+EL, DO+EL), and all three combined (DO+AT+EL).

Algorithm 1 SA for Optimizing Distance

Initialize $d_{Best}, d_{thres}, t_{max}, t_{min}, \alpha$
$t \leftarrow t_{max}$
while $t > t_{min}$ **do**
  for $i = 1, 2, 3, \ldots$ **do**
    $d_i = rand[d_{min}, d_{max}]$
  end for
  Calculate direction to move $\vec{c}$
  Calculate distance $Dis(d_i)$
  $\Delta D = Dis(d_i) - Dis(d_{Best})$
  if $Phase = shrink$ **then**
    if $Dis(d_i) < Dis(d_{Best})$ **then**
      Accept new $d_{Best} = d_i$
    else
      Accept new $d_{Best} = d_i$ *with probability* $1/\exp(\Delta D/t)$
    end if
  end if
  if $Phase = PB$ **then**
    if $d_i < d_{thres}$ *and* $Dis(d_i) > Dis(d_{Best})$ **then**
      Accept new $d_{Best} = d_i$
    else
      Accept new $d_{Best} = d_i$ *with probability* $1/\exp(\Delta D/t)$
    end if
  end if
  Decrease the temperature $t = \alpha * t$
end while

Fig. 3. Simulated Annealing (SA).

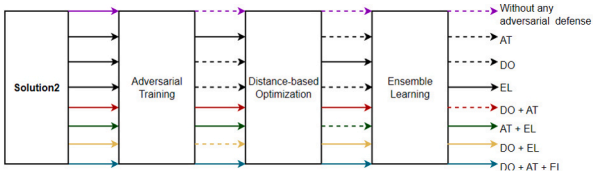

Fig. 4. Combination order.

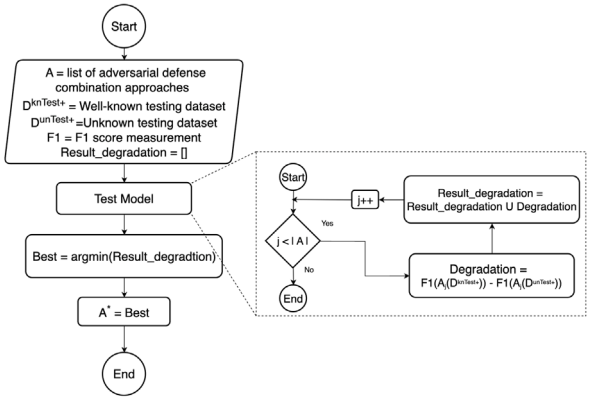

Fig. 5. General framework of our solution.

This comprehensive approach thoroughly evaluates the resilience and effectiveness of various defense strategies against adversarial attacks.

The three combinations of defenses require sequential execution to maintain correct logic. In Fig. 4, solid lines indicate the inclusion of each defense method in the sequence, while dotted lines indicate its exclusion.

In Fig. 5 we present three solutions and their combinations, generating eight models. The inputs include the eight models, a known test data set, and an unknown test data set. By collecting and evaluating all F1 scores, we aim to identify the best F1 score for our IDS.

4.3.2. Adversarial training

In Figs. 15, the input includes three classifiers, an augmented training dataset, and a known adversarial test dataset. Each model is trained

Table 3
Open source and Tools.

Category	Name	Functionality
Library	Sckit-learn	library of Machine learning
	Pytorch	library of Deep learning
	cantools	Encoding and decoding CAN messages and signals, as well as for generating CAN message and signal database.
	python-can	Interfacing with CAN to send and receive messages.
Adversarial Attack	Adversarial Robustness Toolbox	The tool use to attack the model
Simulation	EVπ	Emulate the CAN bus architecture

on the augmented dataset and evaluated on the adversarial test dataset to obtain F1 scores, which are added to a list. The highest F1 score from the list identifies the best adversarial training model.

4.3.3. Ensemble learning

In Figs. 16, the input includes a list of combination model teams, a training dataset, and a testing dataset. For each team, diversity is evaluated by double fault (< 50%) and kappa statistics (40%–80%). The double fault measures the probability that models make the exact incorrect predictions, while kappa statistics assess the consistency between two models. We aim to avoid continuous faults and ensure models have diverse perspectives on the data. Models meeting these thresholds are added to the ensemble teams list. The most frequent ensemble team is identified as the output.

4.3.4. Distance-based optimization

In Figs. 17, the input includes the machine learning model and a known adversarial test dataset. For each model, the distance-based optimization from solution 1 is run, involving finding the optimal distance for shrink and push-back phases. After optimization, the model is tested to determine the best F1 score. The model with the highest F1 score is considered the best Distance-Based Optimization model.

5. Implementation

5.1. Open sources tools and dataset

We used open source tools in our implementation (see Table 3).

Model Library: Scikit-learn for dataset partitioning and evaluation metrics (F1 score, data split), and PyTorch for constructing and training neural networks. These tools supported testing, data preprocessing, and deep learning in our experimental pipeline.

CAN: Cantools for encoding and decoding CAN messages and signals, and python-can for interfacing with the CAN bus and sending messages to EVs. These tools were essential for CAN-based communication in electric vehicles.

Adversarial Attack: The Adversarial Robustness Toolbox (ART) (Nicolae et al., 2018) provides adversarial attack techniques (FGSM, PGD, BIM, etc.) that use model gradient information to generate perturbations maximizing the model's loss function. ART helps evaluate model vulnerability to adversarial examples and develop robust defenses.

Emulation: EVπ (Anon) emulates the CAN bus architecture, allowing model testing in a simulated physical environment for thorough validation and evaluation in real-world scenarios. The complete system is presented in Section 5.3.

5.2. Implementation details

CAN dataset: We generate our dataset independently using the emulated environment provided by EVπ. Each data point conforms to the CAN bus data frame structure, with key features extracted: time interval between CAN messages, ID field, and Data field. Each dataset row contains ten features: time, ID, and eight data fields.

CAN data is captured using candump, converted to CSV, and preprocessed by splitting messages, replacing NaN and empty entries with zero, and labeling instances as benign or malicious. The dataset is split 80/20 for training and testing, respectively, followed by model training and hyperparameter tuning.

ART: ART (Adversarial Robustness Toolbox) uses a dataset and a machine learning model to execute attacks. It employs adversarial dataset generation techniques to manipulate input data and create adversarial examples. After the attack, ART generates an adversarial perturbed dataset containing altered instances to fool the model. This dataset is crucial for evaluating the model's robustness and vulnerability to adversarial attacks.

Distance: the distance-based defense approach consists of two phases: the shrink phase and PB phase. The shrink phase is used to minimize intra-class distance. For this objective, we need a distance and direction. The movement distance d is determined by SA solved in the previous chapter, and the shrink phase movement direction is

$$\overrightarrow{s_i^{j'}} = (v_i^j - c^j). \tag{2}$$

Thus, the expected shrink feature vector is

$$v_i^{j'} = v_i^j + d \times \overrightarrow{s_i^{j'}}. \tag{3}$$

The shrink phase includes a formulation to compute the distance, where

$$\tau_{shrink} = \frac{1}{mn} \sum_{j=1}^{m} \sum_{i=1}^{n} \frac{\|v_i^{j'} - v_i^j\|}{mn} \tag{4}$$

is computed as the mean squared error between feature vectors v_i^j and their corresponding expected feature vectors $v_i^{j'}$, which j is class number and i is the element in j class. On the other hand, the PB phase is used to maximize the distance between classes. To achieve this, we need a formulation to compute the distance and the direction of the center class. The distance d is determined by the SA, and the direction of the center class is

$$\overrightarrow{c^{j'}} = \frac{\sum_{j \neq i} | c^j - c^i |}{K - 1}, \tag{5}$$

determined by the difference between each class center c^j and the other class center point c^i, which j and i are the class number, and K is total class number. So the expected PB feature is

$$v_i^{j'} = v_i^j + d \times \overrightarrow{c^{j'}}, \tag{6}$$

Table 4
LSTM Architecture.

Layer No.	Type	Input size	Output size	Additional details
1	LSTM	(batch, 1)	(batch, 50)	Hidden Size: 50, Layers: 1, Dropout: 0.2
2	LSTM	(batch, 50)	(batch, 50)	Hidden Size: 50, Layers: 1, Dropout: 0.2
3	LSTM	(batch, 50)	(batch, 50)	Hidden Size: 50, Layers: 1, Dropout: 0.2
4	LSTM	(batch, 50)	(batch, 50)	Hidden Size: 50, Layers: 1, Dropout: 0.2
5	Fully-Connected	(batch, 50)	(batch, 2)	Linear Layer (50 → 2)
6	Sigmoid	(batch, 2)	(batch, 2)	Activation

Table 5
DNN Architecture.

Layer No.	Type	Input size	Output size	Additional details
1	Fully-Connected	(batch, 10)	(batch, 10)	Linear Layer (10 → 10)
2	BatchNorm	(batch, 10)	(batch, 10)	BatchNorm1d
3	ReLU	(batch, 10)	(batch, 10)	Activation
4	Dropout (0.1)	(batch, 10)	(batch, 10)	Dropout Layer
5	Fully-Connected	(batch, 10)	(batch, 8)	Linear Layer (10 → 8)
6	BatchNorm	(batch, 8)	(batch, 8)	BatchNorm1d
7	ReLU	(batch, 8)	(batch, 8)	Activation
8	Dropout (0.1)	(batch, 8)	(batch, 8)	Dropout Layer
9	Fully-Connected	(batch, 8)	(batch, 4)	Linear Layer (8 → 4)
10	BatchNorm	(batch, 4)	(batch, 4)	BatchNorm1d
11	ReLU	(batch, 4)	(batch, 4)	Activation
12	Dropout (0.1)	(batch, 4)	(batch, 4)	Dropout Layer
13	Fully-Connected	(batch, 4)	(batch, 2)	Linear Layer (4 → 2)
14	Sigmoid	(batch, 2)	(batch, 2)	Activation

Table 6
CNN Architecture.

Layer No.	Type	Input size	Kernels
1	Convolutional	(batch, 10, L)	5 × 10 × 100
2	ReLU	(batch, 100, L)	–
3	Dropout (0.2)	(batch, 100, L)	–
4	Convolutional	(batch, 100, L)	5 × 100 × 200
5	ReLU	(batch, 200, L)	–
6	Dropout (0.2)	(batch, 200, L)	–
7	Convolutional	(batch, 200, L)	10 × 200 × 400
8	ReLU	(batch, 400, L)	–
9	Max Pool	(batch, 400, L/2)	2
10	Dropout (0.2)	(batch, 400, L/2)	–
11	Fully-Connected	(batch, 400)	400 × 2
12	Sigmoid	(batch, 2)	–

and

$$\tau_{PB} = \frac{1}{mn}\sum_{j=1}^{m}\sum_{i=1}^{n}\frac{\|v_i^{j'} - v_i^{j}\|}{mn} \tag{7}$$

calculates the mean squared error between the expected feature vector $v_i^{j'}$ and the actual class center v_i^{j}, serving as the loss function for model updates.

EL: We form ensemble teams using bagging, selecting models for each team, and using hard voting to combine predictions. The best ensemble team is chosen based on majority decision.

AT: To prepare an extended dataset for adversarial training, we combine the clean dataset with the adversarial attacked dataset, containing both benign and malicious instances. We perform an 80/20 train-test split to ensure models are trained on diverse data, including adversarially perturbed samples.

Evaluation: We compare the performance of each approach using F1 scores. The approach with the highest F1 score against adversarial attacks is considered the best performing method, indicating its robustness in accurately classifying benign and malicious instances.

NN Architecture: We employ three models that are LSTM, DNN, and CNN for implementing our ML-based IDS. The architectures of these models are listed separately in Tables 4, 5, and 6. The model with the highest F1 score is regarded as the most effective ML-based IDS model.

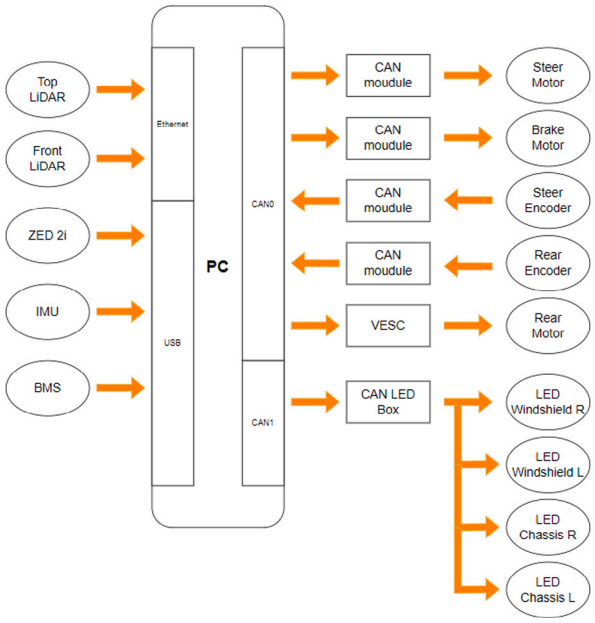

Fig. 6. EVπ environment.

5.3. Testbed EVπ

Fig. 6 depicts EVπ (Anon) as a hardware emulation setup with LiDAR, ZED 2i camera, Inertial Measurement Unit (IMU), Battery Management System (BMS), and controls for steering, braking, rear lights, and LEDs. It mimics a car's functionality but is built on a two-wheeled bicycle structure. The system includes a CAN bus network where CAN 0 handles engine-related messages (steering, braking, rear signals) and CAN 1 manages LEDs. Additionally, LiDAR data is sent through Ethernet, while ZED 2i camera and IMU data use USB connections.

We connect our laptop to EVπ using an OBD-II cable and the python-can library to send CAN messages. These messages are sent to the PC and routed to the correct protocol. For instance, steering system attacks are directed to CAN 0, where the steering motor responds accordingly. Attacks on CAN 0 do not trigger LED signals, posing a risk due to the lack of warning indicators for malicious activity.

In Fig. 7, our attack involves sending malicious CAN messages, directed at the driver, affecting steering, braking, and rear systems. These messages go through the CAN manager to either the human machine interface (HMI) or directly to the driver. Simultaneously, our CAN IDS monitors the network for malicious activity and alerts us via WebSocket if detected.

The interaction of the proposed CAN IDS defense strategies (AT, EL, and DO) is primarily shaped by their roles in offline training and real-time scenarios. Since AT and DO are utilized during the offline model training phase, real-time concerns are not relevant at this stage. Instead, the focus shifts to the performance of EL in real-time scenarios. To address this, we deployed the combined model (AT, EL, and DO) to the EVπ for testing and the model is efficient and capable of accurately predicting CAN messages in the EVπ environment.

Our scenario is designed to simulate attacks originating from external sources. DDoS attacks are common in network environments, while

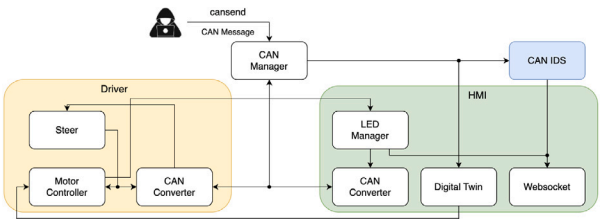

Fig. 7. Attack Path. This figure illustrates the attack path in the CAN network, where malicious CAN messages are injected using the can send tool to compromise the system. The CAN IDS monitors and detects these unauthorized messages to protect the driver and HMI components from potential threats.

Table 7
Parameters configuration.

Model parameters setting	
Learning Rate	0.005 (For DNNs in AT)
Learning Rate	0.001 (For other models and defense methods)
Batch size	2048
Optimizer	Adam
Scheduler	ReduceLROnPlateau
Loss Function	CrossEntropy

spoofing presents a significant threat due to its difficulty in detection. As such, we incorporate both types of attacks to target our vehicle.

During data collection in our scenario, we identified two main attack types: DDoS and Spoofing, which affected the steering, braking, and rear systems. The dataset consists of approximately 1,040,000 CAN messages, divided into 610,000 normal messages and 430,000 malicious messages. These messages are labeled as "0" for benign and "1" for malicious, facilitating the training and testing of our IDS.

6. Experiment results

This chapter is structured into several sections: experiment setting, comparing adversarial attack techniques, evaluating defense methods like distance-based optimization and ensemble approaches, discussing known and unknown attack methods, and comparing the effect of adversarial training with triplet loss. These subsections offer a thorough analysis of adversarial threats and defense strategies.

6.1. Environment setting

Table 7 lists all the parameters used for model training. In the model parameter settings, we specify key parameters such as learning rate, batch size, optimizer, scheduler, and loss function.

6.2. Attack techniques on target models

In Fig. 8, models perform well on normal datasets, with ensemble models enhancing F1 scores through collective decision-making. The ensemble team, comprising DNN, CNN, and LSTM models, showcases the strength of combining diverse architectures for improved performance.

Single vs. Ensemble on Adversarial Dataset

- **The Strongest Adversarial Attack Technique**
  In Fig. 9, the ML-based CAN IDS is subjected to adversarial attacks. The y-axis represents model B attacked by adversary A, while the x-axis displays the selected models and ensembles. Darker cells indicate lower F1 scores, with the overall average dropping to 0.56. The most potent attack, ZOO, reduces the average F1 score to 0.42 due to its optimization-based iterative nature. In contrast, JSMA, the least effective attack, lowers the F1 score to 0.63 using gradient-based methods, though its impact is constrained by the integer limitations of the CAN bus protocol. Despite being the weakest attack, JSMA still significantly impacts the IDS, reducing its F1 score from 0.99 to 0.62.

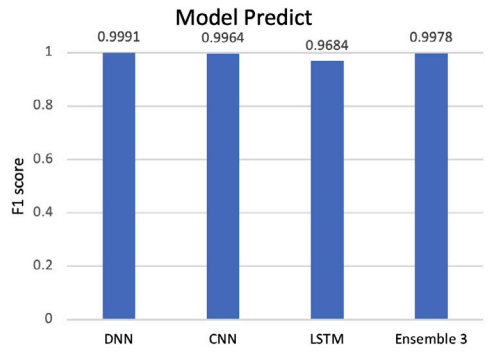

Fig. 8. The result of single and ensemble.

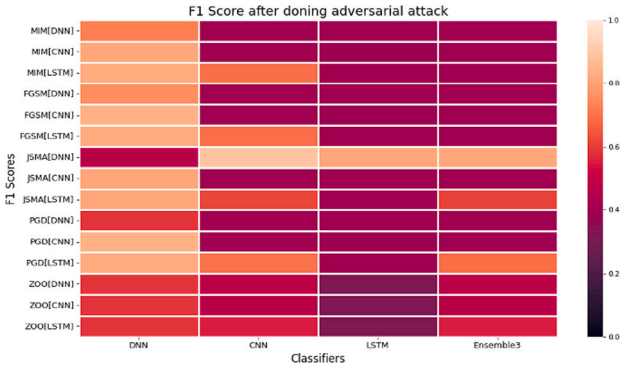

Fig. 9. Adversarial attack result.

- **Classifiers vs. Adversarial Attack Technique**
  Vertically analyzing the heatmap in Fig. 9 reveals all classifiers' vulnerability to adversarial attacks, with F1 scores dropping notably from 0.4 to 0.73. The highest average F1 score of 0.73 by DNN highlights its robustness due to its simpler architecture that provide less information to gradient-based adversarial attacks, making it less susceptible to perturbations. In contrast, LSTM, designed for time series analysis, records the lowest average F1 score of 0.39, as adversarial attacks disrupt both data and temporal aspects, challenging its defense capabilities.

- **Ensemble Team vs. Adversarial Attack**
  The effect of ensemble team's defense against adversarial attacks is not well, with the F1 score decreasing to 0.48 due to poor performance of individual members, LSTM and CNN. The ensemble's success relies on the individual models' quality, impacting its overall performance.

6.3. Distance-based vs. Adversarial training vs. Ensemble

6.3.1. Distance-based optimization

In Fig. 10, Distance-based Optimization enhances the F1 score against ZOO [CNN] from 0.459 to 0.633, showing improvement. Despite its goal of maximizing class separation, it may struggle when ZOO generates data between 0 to 1 units from the original features, as CAN only receives integer values. Consequently, data is mapped to 0 or 1, causing a loss of original characteristics. However, the average F1 score of distance-based optimization is 0.76, showing a 20% improvement

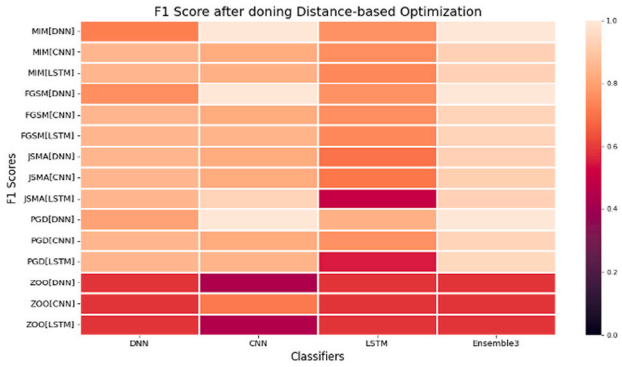

Fig. 10. Distance-based optimization.

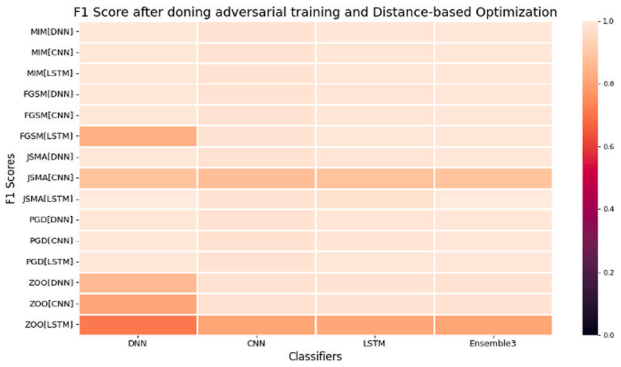

Fig. 12. Distance-based Optimization with AT.

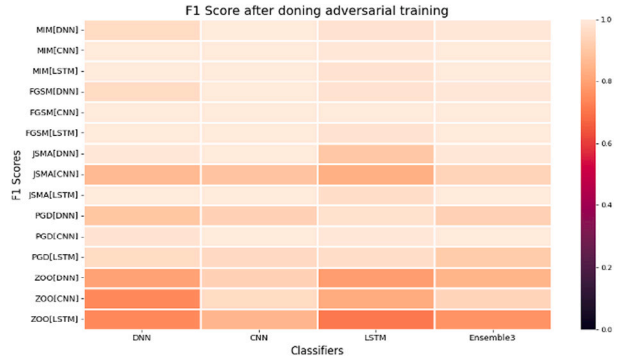

Fig. 11. Adversarial Training (AT).

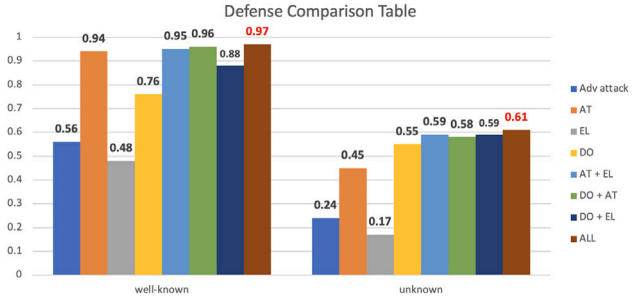

Fig. 13. Comparison with all defense methods on Unknown vs. Well-known.

over the F1 score of the original adversarial attack of 0.56. Combining distance-based optimization with ensemble learning enhances defense by leveraging attack insights for informed decision-making, boosting model robustness against adversarial perturbations, and it can get a better F1 score of 0.88 than distance-based optimization.

6.3.2. Adversarial training

In Fig. 11, adversarial training significantly boosts the F1 score against ZOO[CNN] from 0.32 to 0.84, outperforming distance-based optimization. This improvement is attributed to learning from adversarial examples during training. The average F1 score of adversarial training is 0.94, achieving a 38% improvement over the original adversarial attack. Also, ensemble adversarial training matches individual training in F1 scores, with an ensemble model's 0.95, showing satisfactory results.

6.3.3. Distance-based optimization with adversarial training

Fig. 12 displays the outcomes of combining distance-based optimization with adversarial training, yielding similar results to adversarial training alone. The average F1 score improves to 0.96, surpassing the individual training and achieving a 40% improvement in the original adversarial attack. Notably, the combination of DO+AT+EL achieves the highest F1 score of 0.97. Moreover, ZOO persists as a potent attack, consistent with prior findings, maintaining its strength.

6.4. Adversarial attack: Unknown vs. Well-known

Fig. 13 illustrates the effectiveness of various defense strategies against known and unknown adversarial attacks, with known attacks shown on the left and unknown on the right. For known attacks, Adversarial Training (AT) is the top single defense strategy. Among combined defenses, Distance-based Optimization with Ensemble Learning (DO+EL) shows a modest improvement, but the highest F1 score of 0.97 is achieved with DO+AT+EL. For unknown attacks, AT is not the best single defense; Distance-based Optimization (DO) performs better. Both two-defense combinations yield similar results, with the highest F1 score again achieved by combining all defenses (DO+AT+EL). Overall, DO+AT+EL is the most effective strategy for both attack types.

6.5. Compare defense effect of adversarial training with triplet loss

Adversarial training with triplet loss (Li et al., 2019) uses adversarial examples as the anchor, original examples as the positive node, and examples from other classes as the negative node. The goal is to maximize the distance between the anchor and the negative node while minimizing the distance between the anchor and the positive node.

Adversarial training with triplet loss aims to achieve similar goals to our distance-based optimization method using a triplet loss function and adversarial examples. However, our approach yields better results. In known adversarial attacks, the F1 score of distance-based optimization is 12% higher than AT+Triplet, and 8% higher in unknown attacks. Furthermore, the DO+AT method scores 32% higher in known attacks and 11% higher in unknown attacks compared to AT+Triplet. Finally, our DO+AT+EL method improves the F1 score by 33% in known attacks and 14% in unknown attacks over AT+Triplet. These findings highlight that our DO+AT+EL method is the most effective defense.

Our method outperforms adversarial training with triplet loss for several reasons. Firstly, adversarial training with triplet loss focuses solely on relative distances between the anchor, positive, and negative samples, neglecting absolute distance. Additionally, it does not explicitly consider intra-class distance. In contrast, our method uses shrink to compute absolute intra-class distances and PB for absolute inter-class distances. We also employ simulated annealing to optimize distance adjustment, contributing to our approach's superior performance over adversarial training with triplet loss (see Fig. 14).

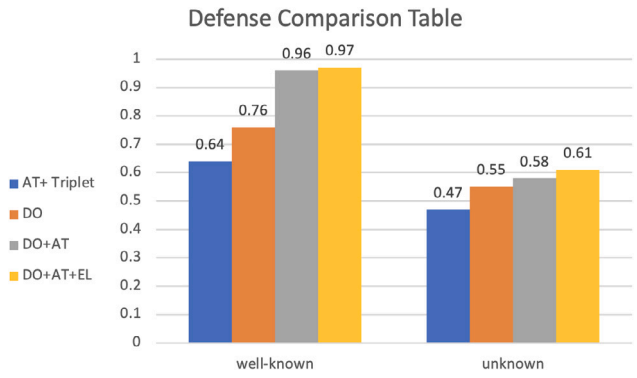

Fig. 14. AT+triplet loss vs. DO vs. DO+AT vs. DO+AT+EL.

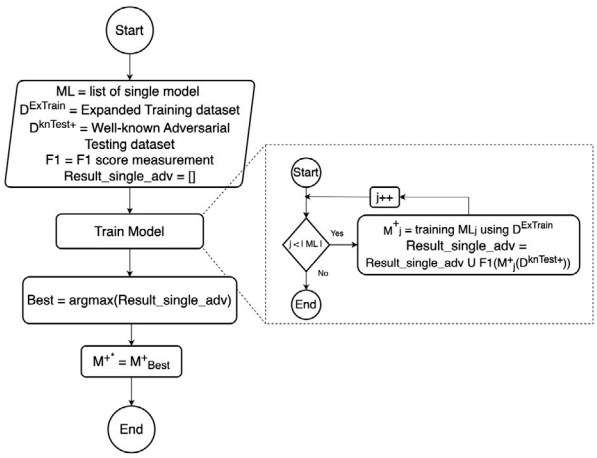

Fig. 15. Adversarial training in our solution.

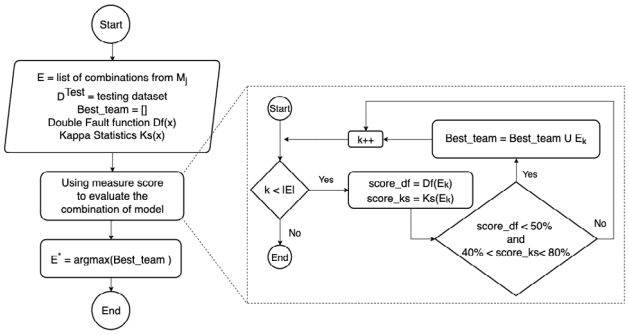

Fig. 16. Ensemble learning in our solution.

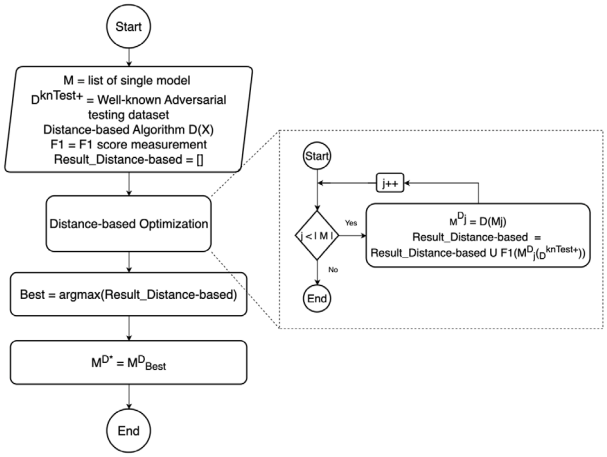

Fig. 17. Distance-based optimization in our solution.

7. Conclusions and future work

This chapter summarizes the paper's findings and suggests future research directions to enhance understanding and defense in the CAN domain.

DO+AT+EL is better than AT+triplet loss: Our study shows that our method outperforms adversarial training with triplet loss. For known attacks, our method DO+AT+EL achieves an F1 score of 0.97, versus 0.64 for adversarial training, a 33% improvement. For unknown attacks, our method scores 0.61, compared to 0.47 for adversarial training, a 14% improvement. This is because our method calculates absolute intra-distance and inter-distance, using an optimization algorithm to determine the optimal movement distance, unlike adversarial training, which only considers relative inter-class distance. Overall, our method excels against both known and unknown attacks.

Combining DO, AT, and EL yields the best results: Adversarial defense methods varied in effectiveness. Adversarial training achieved an F1 score of 0.94 against known attacks but only 0.45 against unknown ones due to its focus on known patterns. Ensemble learning faced bias issues, scoring just 0.17 against unknown attacks because all models were affected, resulting in poor voting. Distance-based optimization, not relying on adversarial examples, performed well, scoring 0.76 and 0.55 against known and unknown attacks, respectively. Combining all three defenses yielded the best results, with F1 scores of 0.97 and 0.61, showing the benefits of multiple strategies.

ZOO is the strongest adversarial attack & CNN is the most resilient model: The ZOO attack proved the strongest, dropping the F1 score to 0.42. Despite this, the DNN model held a robust 0.73 F1 score, and the CNN model was the most resilient, consistently performing well against adversarial threats. This underscores the need to understand model vulnerabilities and resilience to develop robust security measures.

Future work: Future research will expand the range of adversarial attacks and test defense strategies in actual vehicles instead of emulation environments like EVπ. This real-world testing aims to enhance automotive cybersecurity by providing practical insights into our methods' effectiveness. Additionally, considering actual vehicles' limited resources, it will be crucial to assess the feasibility of implementing our defense architecture in such constrained environments and optimize resource usage.

CRediT authorship contribution statement

Ying-Dar Lin: Writing – review & editing, Supervision, Project administration, Conceptualization. **Wei-Hsiang Chan:** Writing – original draft, Software, Methodology, Data curation. **Yuan-Cheng Lai:** Supervision, Project administration, Data curation, Conceptualization. **Chia-Mu Yu:** Writing – review & editing, Validation, Methodology. **Yu-Sung Wu:** Validation, Supervision, Formal analysis. **Wei-Bin Lee:** Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Chia-Mu Yu reports financial support was provided by Hon Hai Research Institute. Ying-Dar Lin reports financial support was provided by Hon Hai Research Institute. Wei-Hsiang Chan reports financial support was provided by Hon Hai Research Institute. Yu-Sung Wu reports financial support was provided by Hon Hai Research Institute. Wei-Bin Lee reports financial support was provided by Hon Hai Research Institute. If there are other authors, they declare that they have no known

competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was partially supported by Hon Hai Research Institute.

Appendix

See Figs. 15–17.

Data availability

Data will be made available on request.

References

Anon, EVπ architecture. [Online]. Available: https://www.csltaipeitech.com/ummanned-vehicle.

Anthi, E., Williams, L., Rhode, M., Burnap, P., Wedgbury, A., 2021. Adversarial attacks on machine learning cybersecurity defences in industrial control systems. J. Inf. Secur. Appl. 58, 102717.

Ashraf, S., Ahmed, T., 2020. Sagacious intrusion detection strategy in sensor network. In: 2020 International Conference on UK-China Emerging Technologies. UCET, IEEE, pp. 1–4.

Bertsimas, D., Tsitsiklis, J., 1993. Simulated annealing. Statist. Sci. 8 (1), 10–15.

Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., Hsieh, C.-J., 2017. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. pp. 15–26.

Chien, J.-T., Chen, Y.-A., 2024. Towards a unified view of adversarial training: A contrastive perspective. In: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 5365–5369.

Demontis, A., Melis, M., Pintor, M., Jagielski, M., Biggio, B., Oprea, A., Nita-Rotaru, C., Roli, F., 2019. Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks. In: 28th USENIX Security Symposium (USENIX Security 19). pp. 321–338.

Deng, Y., Mu, T., 2024. Understanding and improving ensemble adversarial defense. Adv. Neural Inf. Process. Syst. 36.

Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J., 2018. Boosting adversarial attacks with momentum. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9185–9193.

Goodfellow, I.J., Shlens, J., Szegedy, C., 2014. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

Li, P., Yi, J., Zhou, B., Zhang, L., 2019. Improving the robustness of deep neural networks via adversarial training with triplet loss. arXiv preprint arXiv:1905.11713.

Lin, Y.-D., Pratama, J.-H., Sudyana, D., Lai, Y.-C., Hwang, R.-H., Lin, P.-C., Lin, H.-Y., Lee, W.-B., Chiang, C.-K., 2022. ELAT: Ensemble learning with adversarial training in defending against evaded intrusions. J. Inf. Secur. Appl. 71, 103348.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A., 2017. Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083.

Mao, C., Zhong, Z., Yang, J., Vondrick, C., Ray, B., 2019. Metric learning for adversarial robustness. Adv. Neural Inf. Process. Syst. 32.

Martins, N., Cruz, J.M., Cruz, T., Abreu, P.H., 2020. Adversarial machine learning applied to intrusion and malware scenarios: a systematic review. IEEE Access 8, 35403–35419.

Miller, C., Valasek, C., 2013. Adventures in automotive networks and control units. Def Con 21 (260–264), 15–31.

Miyato, T., Maeda, S.-i., Koyama, M., Nakae, K., Ishii, S., 2015. Distributional smoothing with virtual adversarial training. arXiv preprint arXiv:1507.00677.

Moosavi-Dezfooli, S.-M., Fawzi, A., Frossard, P., 2016. Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2574–2582.

Mustafa, A., Khan, S.H., Hayat, M., Goecke, R., Shen, J., Shao, L., 2020. Deeply supervised discriminative learning for adversarial defense. IEEE Trans. Pattern Anal. Mach. Intell. 43 (9), 3154–3166.

Natale, M.D., Zeng, H., Giusto, P., Ghosal, A., 2012. Understanding and Using the Controller Area Network Communication Protocol: Theory and Practice. Springer Publishing Company, Incorporated.

Nicolae, M.-I., Sinn, M., Tran, M.N., Buesser, B., Rawat, A., Wistuba, M., Zantedeschi, V., Baracaldo, N., Chen, B., Ludwig, H., et al., 2018. Adversarial robustness toolbox v1. 0.0. arXiv preprint arXiv:1807.01069.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A., 2016. The limitations of deep learning in adversarial settings. In: 2016 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, pp. 372–387.

Seo, S., Lee, Y., Kang, P., 2023. Cost-free adversarial defense: Distance-based optimization for model robustness without adversarial training. Comput. Vis. Image Underst. 227, 103599.

Strauss, T., Hanselmann, M., Junginger, A., Ulmer, H., 2017. Ensemble methods as a defense to adversarial perturbations against deep neural networks. arXiv preprint arXiv:1709.03423.

Wang, Z., 2018. Deep learning-based intrusion detection with adversaries. IEEE Access 6, 38367–38384.

Wang, C.-W., Lin, Y.-D., Lai, Y.-C., Wu, Y., Yu, C.-M., Chen, Y.-S., Lee, W.-B., 2023. In-parallel defense in ML-based CAN IDS: Detect-and-denoise, adversarial training, and ensemble learning.

Wen, Y., Zhang, K., Li, Z., Qiao, Y., 2016. A discriminative feature learning approach for deep face recognition. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, the Netherlands, October 11–14, 2016, Proceedings, Part VII 14. Springer, pp. 499–515.

Ying-Dar Lin is a Chair Professor of computer science at National Yang Ming Chiao Tung University (NYCU), Taiwan. He received his Ph.D. in computer science from the University of California at Los Angeles (UCLA) in 1993. He was a visiting scholar at Cisco Systems in San Jose during 2007–2008, CEO at Telecom Technology Center, Taiwan, during 2010–2011, and Vice President of National Applied Research Labs (NARLabs), Taiwan, during 2017–2018. He cofounded L7 Networks Inc. in 2002 and O'Prueba Inc. in 2018. His research interests include cybersecurity, wireless communications, network softwarization, and machine learning for communications. He is an IEEE Fellow (class of 2013). He has served or is serving on the editorial boards of several IEEE journals and magazines, including Editor-in-Chief of IEEE Communications Surveys and Tutorials (COMST, 2017–2020).

Wei-Hsiang Chan received his M.S. degree at Institute of Network Engineering of National Yang Ming Chiao Tung University (NYCU) in 2024. He was an associate researcher at High Speed Network Lab, NYCU, in 2022–2024. His research interests include cybersecurity and machine learning.

Yuan-Cheng Lai received his Ph.D. degree in the Department of Computer and Information Science from National Chiao Tung University in 1997. He joined the faculty of the Department of Information Management at National Taiwan University of Science and Technology in August 2001 and has been a distinguished professor since June 2012. His research interests include performance analysis, software-defined networking, wireless networks, and IoT security.

Chia-Mu Yu is a Associate Professor of Electronics and Electrical Engineering at National Yang Ming Chiao Tung University (NYCU). He received his Ph.D. degree in the Graduate Institute of Electrical Engineering from National Taiwan University (NTU) in 2013. His research interests include AI Safety/Security/Robustness, Computer and Network Security, Data Privacy and Anonymization.

Yu-Sung Wu is a Professor of Computer Science at National Yang Ming Chiao Tung University (NYCU). He received a Bachelor's degree from National Tsing Hua University (NTHU) in 2002 and a Ph.D. in Electrical and Computer Engineering from Purdue University in 2009. His research interests are in security, dependability, and systems. He is currently the director of the Graduate Degree Program of Cybersecurity and the deputy director of the Center for Education and Research in Cybersecurity at NYCU.

Wei-Bin Lee received his Ph.D. degree from National Chung Cheng University in 1997. Dr. Lee served as a professor in the Department of Information Engineering & Computer Science at Feng Chia University. He was also a visiting professor at both Carnegie Mellon University in the USA and University of British Columbia in Canada. Since 2021, he has served as the CEO of HonHai Research Institute as well as the director of the information security research center. Before joining the Foxconn group, the CEO WeiBin Lee held important positions in Taipei City Government, Taipei Fubon Bank, Fubon Financial Holdings, and Feng Chia University. His research experiences fall in network security, cryptography, digital rights management, and privacy/security management and governance.