A Robust Ensemble Malware Detector Against Powerful Adversaries

Shangyuan Zhuang[1,2], Wei Zhang[3], Fengqi Liu[3], Jiyan Sun[1] (✉), Yinlong Liu[1,2], Liru Geng[1,2], and Wei Ma[1]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{zhuangshangyuan, sunjiyan, liuyinlong, gengliru, mawei}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] Henan Jiuyu EPRI Electric Power Technology Co., LTD., Henan, China
zw15272@163.com, liu_fengqi@126.com

Abstract. Considering the vulnerability of machine learning to adversarial attacks, the state-of-the-art malware detectors are designed with appropriate ensembles. However, most ensemble detectors are developed based on unreasonable and weak threat assumptions, which do not match the characteristics of real-world adversaries with powerful adaptive mixed capabilities. Such detectors will unavoidably suffer severe failures in practical deployment. To this end, we build two realistic powerful adversary models and propose *NashAE* as a robust malware detector based on a novel Game-theory-enabled ensemble adversarial training approach against them. Specifically, *NashAE* establishes a Minimax Game where the adversary and detector compete on opposing targets. By solving the Nash equilibrium of the game, *NashAE* can obtain the optimal ensemble adversarial training strategy under adversaries' constantly adaptive attacks. Since the game has no closed-form solution, we further develop a simplified solution scheme based on Bayesian optimization to find the approximate Nash equilibrium of the game. We conduct comprehensive experiments with 10 baseline detection models on 2 malware datasets. Experimental results show that *NashAE* can achieve a stable detection rate of 58% higher than advanced methods after only 15 iterations against the most powerful adversaries.

Keywords: Malware detector · Adversarial defense · Nash equilibrium solution · Game theory · Ensemble learning · Network security.

1 Introduction

The deep integration of 5G and satellite networks as communication carriers for critical infrastructure (e.g., new distribution grid systems) has expanded the propagation radius of malware from terrestrial systems to space networks. One of the mainstream solutions identifies malware based on network traffic since traffic can capture all behaviors in the network [15,17,21]. By analyzing whether the traffic is benign or malicious based on machine learning techniques, malware can be detected. Although these detectors have achieved certain success in

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_2

practice, they are vulnerable to adversarial attacks [10]. Through adding carefully designed adversarial perturbations to malware traffic, the adversarial attack can trick detectors to identify malware traffic as benign. Then these adversarial malware traffic will escape the detector's detection, so that the corresponding malware can commit malicious activities without any hindrance.

To robustly detect malware traffic even in the presence of adversarial perturbations, detectors based on ensemble learning are considered a promising solution, since adversaries have to generate adversarial malware traffic that can confront all base models [8, 14]. Nonetheless, the transferability of adversarial samples renders simple ensemble detectors still unreliable [20]. To further protect against those transfer-based attacks, ensemble adversarial training approaches have been introduced [7, 9, 16, 18], which allow the detector to pre-learn the features of the adversarial traffic in the ensemble training phase. However, when facing real-world adversaries, these advanced detectors are still indefensible because their ensemble strategies are usually developed based on unrealistic and weakness adversarial attack assumptions. More precisely, the characteristics of adversarial traffic learned by them do not match the real adversaries.

Firstly, existing ensemble adversarial training strategies are usually tailored to individual adversarial perturbation types, whereas real-world adversaries are much more sophisticated and powerful. For one thing, in reality, there are multiple adversaries from different organizations with different goals to generate various types of adversarial perturbations. For another, adversaries will have varied attack strategies thanks to open-source adversarial attack tool [2]. Aiming to achieve strong evasion performance, powerful adversaries tend to launch hybrid attacks that mix multiple attack strategies. Further, there are also more powerful adversaries even can adaptively adjust their hybrid attack strategies based on the detector's detection results. With this regard, in the ensemble adversarial training phase, practical detectors have to consider adversaries with mixed attack schemes or even adaptive mix attack schemes, rather than the current simple individual attack schemes.

Secondly, existing ensemble adversarial training strategies usually neglect the traffic-space constraints and knowledge limitation of practical adversarial attacks [3]. For one thing, after adding adversarial perturbations, adversarial malware traffic must still follow the specific semantics of communication protocols. Only such adversarial traffic is effective. However, existing detectors directly learn all adversarial traffic features, causing not only limited improvement in robustness but also leading to detection performance degradation and sometimes even creating more vulnerabilities for detectors. For another, adversaries have no access to their target detectors in reality, while most ensemble detectors learn features of the adversarial samples generated based on the white-box assumptions [13, 19]. It is clear that such adversarial traffic features are not fully consistent with adversarial traffic generated in real-world settings. This further limits the robustness improvement by the ensemble adversarial training.

To address the above challenges, we build two realistic threat models and propose *NashAE* as a robust ensemble malware detector based on a novel Game-

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_2

theory-enabled ensemble adversarial training. By building and solving a Minimax Game where the adversary and detector compete based on the opposite target in the ensemble adversarial training phase, *NashAE* automatically updates its ensemble strategy according to adversaries' constantly adaptive attack scheme. The main contributions of our work are summarized as follows:

- **NashAE Detector against Real-World Powerful Adversaries.** We design *NashAE* ensemble detector which is based on game theory to address the challenge of practical adversaries' adaptive mixed adversarial attacks. By establishing a Minimax Game, *NashAE* simulates the process that the adversaries continuously adjust their attack scheme and detector continuously selects its optimal ensemble strategy. Based on this more realistic dynamic process, *NashAE* has greater potential to defend against the admixture and changing nature of adversarial attackers in real world.
- **Practical Threat Model Building.** We build two practical threat models, M-TBA and AM-TBA, that more closely reflect real-world powerful attacks. Under the M-TBA model, adversaries launch black-box attacks that are mixed and conform to the traffic-space constraints. Under the AM-TBA model, adversaries launch black-box attacks that can adjust their mixed schemes with a certain adaptive ability and conform to the traffic-space constraints. In particular, via a well-designed remapping function, the adversarial traffic can strictly satisfy the traffic-space constraints.
- **Simplified Solution to Nash Equilibrium.** Since Minimax Game has no closed solution, typical gradient-based algorithms will fail to solve the Nash equilibrium that ends the game. To address this problem and achieve rapid response in security field, we provide a simple solution method with relatively less calculation times. Specifically, we convert the Minimax Game into two related Markov Decision Processes (MDPs) and solve them by Bayesian optimization. The solutions of MDPs can be considered as the approximate Nash equilibrium, which is used to obtain optimal ensemble strategy.
- **Comprehensive evaluations of NashAE.** We evaluate our realistic attack and defense methods with 2 encrypted malware traffic datasets using 10 detection strategies in both M-TBA model and AM-TBA model. Experimental results demonstrate that *NashAE* can defend against realistic attacks with stable high performance after only 15 iterations, outperforming other advanced methods by at least 40%. Besides, we verify the feasibility of black-box attacking methods based on the transferability of adversarial perturbations according to our experiments.

2 Related Works

2.1 Adversarial Attack to Malware Detectors

Machine learning techniques have been widely used in malware traffic detection [15,17,21]. Thanks to the powerful classification capability of machine learning, these detectors achieve certain success in practice. Unfortunately, Rigaki *et*

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_2

al. [13] demonstrate that traffic detectors based on machine learning are vulnerable to adversarial attacks. By adding adversarial perturbations that maximize the loss function of target detectors, adversarial attacks can cause detector misclassification. Considering the widespread encryption protocols such as HTTPS, Novo *et al.* [10] implement adversarial attacks take account into encryption traffic. Since attackers hardly have knowledge of target detectors, transfer-based adversarial attacks [5, 12, 24] are proposed to perform black-box attacks. These attacks first create substitute models of their target detectors, and then generate adversarial malware traffic based on the fully accessible substitute models. Benefiting from the transferability of adversarial examples, this adversarial malware traffic has the potential to trick target malware detectors.

2.2 Ensemble detectors against adversarial attacks

Malware detectors based on ensemble learning are considered a potential way to defend against adversarial attacks [7]. Nevertheless, some works demonstrate that detectors based on ensemble learning are still insufficient to robustly defend against adversarial perturbations [22, 23]. For example, Zhang *et al.* [23] show that discrete-valued tree ensemble models can be easily attacked by adversarial examples generated according to the models decision output.

To further improve the robustness of the ensemble model, ensemble adversarial training strategies [16] are proposed. Specifically, Tramer *et al.* [16] first augment training data with adversarial perturbations transferred from other models to increase the robustness of ensemble detectors. Following them, also some studies show robust defense ways for adversarial attacks via ensemble adversarial learning [9,18]. However, due to unrealistic assumptions of adversaries, these detector ensemble strategies are fixed and learn some features of adversarial traffic that do not satisfy the traffic-space constraints. This inevitably invalidates their robust defense strategies against realistically powerful adversaries. Further, Shu *et al.* [14] propose the Omni ensembles models, whose hyperparameters are controlled to make the attacker's target model far away. Based on their idea of hyper-parametric control, Omni has a certain ability to defend against powerful adversaries with mixed strategies (i.e., the M-TBA threat model). But it will be vulnerable to more powerful adversaries that can dynamically and adaptively adjust their mixed attack strategies (i.e., the AM-TBA threat model).

3 Overview of NashAE Ensemble Detector

In this section, we will briefly introduce the overall structure of our NashAE ensemble detector, which is capable of robustly defending against the admixture and adaptive nature of adversarial attackers in the real world.

As shown in Fig. 1, *NashAE* enhances detection robustness through a novel ensemble adversarial training approach. Considering that the most powerful adversaries have mixed attack schemes and adapt their attack schemes according to attack performance and defense strategies, *NashAE* is adversarially trained

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_2

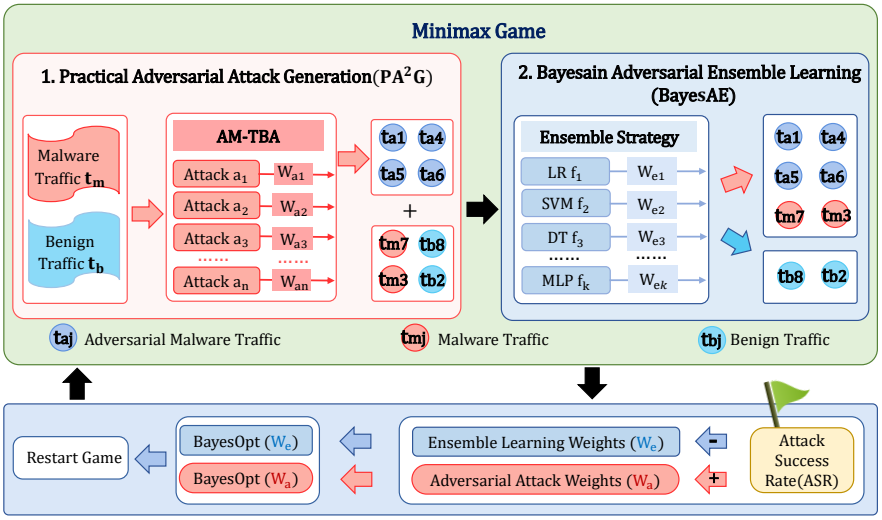


Fig. 1: The overall structure of *NashAE* ensemble adversarial training approach.

with a dynamic ensemble strategy. To describe the dynamic process in which the detector adjusts its ensemble strategy in line with adversaries' constantly adaptive attack schemes, *NashAE* establishes a Minimax Game. The two players of the game are practical adversarial attack generation (PA^2G) and Bayesian adversarial ensemble learning (BayesAE).

Among them, PA^2G simulates a realistic adversary with mixed adaptive attack scheme, and BayesAE simulates an ensemble detector with dynamic ensemble strategies. Before the game starts, PA^2G has a range of base adversarial attack resources $a_1, a_2, ..., a_n$ that can generate a series of adversarial malware traffic t_a from the malicious traffic t_m. BayesAE has a range of base malware detection resources $f_1, f_2, ..., f_k$ that aim to detect all malware traffic. The two players align their strategies toward opposite targets during the game. PA^2G tries to find a series of base adversarial attack weights $\mathbf{w_a} = \{w_{a1}, w_{a2}, ..., w_{an}\}$ that maximum attack success rate (ASR), so that the realistic adversary can best fool the current ensemble detector. Meanwhile, BayesAE tries to adjust a series of ensemble weights $\mathbf{w_e} = \{w_{e1}, w_{e2}, ..., w_{en}\}$ that minimize the ASR, so that the ensemble detectors can best robust against the current attack. Let $g(\cdot)$ denote the objective ASR that the two players want to maximize/minimize, then the minimax game can be described as:

$$\min_{w_{ei} \in \mathbf{w}_e} \max_{w_{ai} \in \mathbf{w}_a} g(\mathbf{w}_e, \mathbf{w}_a) \tag{1}$$

Since they compete on opposite directions of ASR, the game will reach a Nash equilibrium, which is the solution of the most robust ensemble detector. Detailed strategies of the two players are described below.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_2

4 Practical Adversarial Attack Generation (PA^2G)

As one player of the Minimax Game, PA^2G aims to generate a series of optimal adversarial malware traffic that meets the characteristics of real-world powerful adversaries, which can be used as part of samples for ensemble adversarial training. The detailed design principle of PA^2G will be elaborated in this section.

4.1 Threat Model

Since it is difficult for adversaries to obtain detailed knowledge of their target detector in reality, adversarial attacks are usually performed under black-box scenarios. Considering that the network will automatically drop non-compliant packets, only adversarial malware traffic that satisfies the traffic space constraints is effective. For a high attack success rate, real-world adversaries usually launch attacks that mix multiple schemes. Depending on whether the adversary can adaptively adjust their attack mixed scheme, we define the following 2 threat models, aiming to match the characteristics of a realistic and powerful adversary.

1. ***M-TBA***: As a mixed traffic-space black-box attack, M-TBA has no adaptive capability. In this case, adversaries launch attacks based on the average mixed scheme, rather than mixed schemes with different weights. Thus it can be directly defended without establishing the Minimax Game.
2. ***AM-TBA***: As an adaptive mixed traffic-space black-box attack, AM-TBA can constantly adapt its mixed scheme according to the current attack performance and defense strategies. In this dynamic case, we have to establish a game to solve the most robust detector ensemble strategy.

To obtain an optimal detection ensemble strategy with the upper robustness performance, PA^2G simulates the most powerful adversaries AM-TBA. It is clear that the detection strategy obtained from AM-TBA threat model also can robustly resist the M-TBA threat model. Detailed principles are as follows:

· **Principle of Mixed Attack** In the case of M-TBA, the adversary has multiple adversarial attack methods $a_1, a_2, ..., a_n$ with a series of fixed mixture weight factor $w_{a1}, w_{a2}, ..., w_{an}$, aiming to evade the detection of ensemble detector $\mathbf{f}$ that integrates multiple base detectors $f_1, f_2, ..., f_k$ with a certain strategy.

· **Principle of Adaptive Mixed Attack** In the case of AM-TBA, we solve adversary's optimal attack mixed scheme by establishing the Minimax Game. As shown in Fig. 1, there are multiple adversarial attack methods $a_1, a_2, ..., a_n$ in PA^2G. To effectively attack the ensemble detector $\mathbf{f}$ that integrates multiple base detectors $f_1, f_2, ..., f_k$ with a certain strategy, the adversary will mix $a_1, a_2, ..., a_n$ by a series of weight factor $w_{a1}, w_{a2}, ..., w_{an}$.

· **Principle of Black-Box Attack:** Since we consider black-box scenarios that adversary has no knowledge of $\mathbf{f}$, adversaries cannot directly find appropriate adversarial perturbations δ according to $\mathbf{f}$. To this end, they first locally train

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_2

an ensemble model f' as the substitute model of target detector $\mathbf{f}$. Then δ can be generated according to substitute model f', which is considered effective perturbation to $\mathbf{f}$. By adding δ to malware traffic t_m, PA^2G can obtain adversarial malware traffic t_a that has the potential to trick the current $\mathbf{f}$.

· **Problem Modeling:** For a successful attack in practice, t_a needs to be deceptive (i.e. misclassified as t_m by the detector based on its extracted feature $\phi(t_a)$), and satisfy traffic-space constraints Θ. This problem can be described as:

$$\text{argmin}_{t_a} f'\left(\phi'(t_a)\right) + \lambda \mathcal{L}(t_a, t_m) \qquad \text{s.t. } t_a \in \Theta(t_m) \tag{2}$$

where $\phi'(t_a)$ represents the extracted feature of adversarial malware traffic t_a, λ indicates a parameter that determines the feature transformation cost from f' to $\mathbf{f}$, $\mathcal{L}$ is the loss function of $\mathbf{f}$, and $\Theta(t_m)$ represents the traffic-space constraints with the same malicious function and communication protocol as t_m.

· **Adversarial Malware Traffic t_a Generation:** To generate adversarial malware traffic t_a satisfying the traffic-space constraints, we take the following two steps. First, we generate adversarial feature x_a from a clean input x_m in feature space. Second, we design a remapping function $\mathcal{M}$ to project x_a to obtain the ultimate traffic-space adversarial traffic t_a, which is satisfying the traffic-space constraints. Details will be introduced in Sec. 4.2 and Sec. 4.3, respectively.

4.2 Generating Adversarial Features x_a

To find the optimal adversarial perturbation δ that can trick the substitute model f' to identify adversarial malware feature $x_m + \delta$ as a benign feature x_b, we need to maximize the loss function $\mathcal{L}'$ of f'. Let y represents the output label by f' according to the input features, the optimization problem for optimal δ can be expressed as:

$$max\, \mathcal{L}'(f'(x_m + \delta), y) \tag{3}$$

To solve the above optimization problem, a common method is to find δ along the gradient ascent direction of loss function $\mathcal{L}'$. Unfortunately, there are non-differentiable tree models in the ensemble substitute model f' trained locally by the adversary. This will cause the classical gradient ascent method to not work. Inspired by [6], we consider evaluating the gradient of loss function based on Natural Evolutionary Strategies (NES). Then the gradient ascent algorithm can be performed according to the estimated gradients.

Specifically, loss function $\mathcal{L}'(\theta)$ can be approximated as $\mathcal{L}'(\theta + \xi)$ through adding a series of search noise ξ_i with distribution $\pi(\theta|x_m)$. Rather than directly maximizing $\mathcal{L}'(\theta)$, NES maximizes the expected value of the loss function under the search distribution. By sampling n points of ξ_i values that obey $\xi_i \sim \mathcal{N}(0, I)$, the gradient G can be estimated by:

$$G \approx \nabla \mathbb{E}[\mathcal{L}(\xi)] = \frac{1}{\sigma n} \sum_{i=1}^{n} \xi_i \mathcal{L}\left(\theta + \sigma \delta_i\right) \tag{4}$$

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_2

Based on the estimated gradient G of substitute model f' loss function, attackers can iteratively find optimal perturbation δ along the direction of gradient ascent. Then the adversarial features x_a can be generated by $x_a = x_m + \delta$. Thanks to the transferability of adversarial attacks, x_a can trick the target detector $\mathbf{f}$ with certain effectiveness.

4.3 Projecting Adversarial Features x_a to t_a in Traffic Space

Adversarial traffic generated in practice should strictly satisfy the traffic-space constraints, that is, follow the communication protocol and preserve the malicious functionality of malware. To enforce these domain constraints, we use the remapping function $\mathcal{M}$ to adjust the perturbed features. There are some inherent characteristics of network traffic, such as flow-based features cannot be changed by attackers, some features depend on other features. For a more reasonable feature remapping, we group traffic features into the following four categories:

1. **Features with unchangeable value:** To maintain the malicious function of the original malware, some features shouldn't be changed because they are extracted from backward packets (i.e., victim packets).
2. **Features with zero value:** To prevent modified adversarial traffic from being directly dropped by the network, the modified features have to maintain the correctness of the network protocol. This requires certain features to have eigenvalues of 0 (e.g. the value of 'TCP flag').
3. **Features with valid interval boundaries:** Based on our analysis of different types of malware traffic, we found that there are upper and lower bounds on the value of each feature for different malware traffic, i.e., these feature values have validity intervals. In order to maintain the original malicious function, it is necessary to ensure that the modified feature values remain within the valid interval boundaries.
4. **Features depending on other features:** Since existing feature extractors usually extract statistical features of encrypted traffic, there is a correspondence between some feature values computed based on the same attributes. That is, if one feature value is modified, the feature values of other related features need to be changed accordingly.

We design remapping function $\mathcal{M}$ to restrict the adversarial malware traffic generated by modifying the traffic features to satisfy the above four types of constraints. The remapping function $\mathcal{M}$ mainly takes three technical solutions, which are masking, clip, and equation constraints. For the features with unchangeable values, $\mathcal{M}$ uses the masking technique to refuse adversaries to perturb these features when generating adversarial samples. For the features with zero value and the features with valid interval boundaries, $\mathcal{M}$ uses the clip technique to ensure that the perturbed feature values are 0 or within bounds. For the features depending on other features, $\mathcal{M}$ uses equation constraints to maintain the correlation between the feature values. According to $\mathcal{M}$, we can map the adversarial malware features x_a to the adversarial malware traffic t_a that satisfies the traffic-space constraints. Obviously, t_a has the potential to evade detection by $\mathbf{f}$, and will not be dropped by the network since network can understand it.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_2

5 Bayesian Adversarial Ensemble Learning (BayesAE)

As the other player of Minimax Game, BayesAE aims to obtain a more robust ensemble detector with a dynamic ensemble strategy. The ensemble strategy when the game reaches Nash equilibrium is capable to defend against the current real-world adaptive mixed adversarial attacks. To solve the Nash equilibrium with little computation times, BayesAE provides a simplified solution. The detailed design principle of BayesAE will be elaborated in this section.

5.1 Dynamic Ensemble Strategy of Detector

To deal with the adaptively mixed attack of real-world adversaries, BayesAE constantly adjusts the ensemble strategy of *NashAE* according to current adversaries' attack schemes. This process is described as a Minimax Game between PA^2G and BayesAE, which is shown in Fig. 1. Specifically, the adaptive adversary constantly updates its mixed attack scheme that is composed of multiple attacks $a_1, a_2, ..., a_n$ with different weights $\mathbf{w}_a = w_{a1}, w_{a2}, ..., w_{an}$. Each update is performed with the goal of maximizing its attack success rate (ASR). Aiming to defend against every mixed attack launched by the adversary, *NashAE* constantly reconfigures a series of ensemble weights $\mathbf{w}_e = w_{e1}, w_{e2}, ..., w_{en}$ of multiple detector models. Each reconfiguration is performed with the goal of minimizing the updated ASR.

In each round of the game, PA^2G and BayesAE evaluate ASR under the current strategies. To maximize and minimize ASR by the two players, we update the ensemble learning weights $\mathbf{w}_e$ of ensemble strategy and the adversarial attack weights $\mathbf{w}_a$ by Bayesian optimization. After multiple rounds of fictitious play, the ensemble strategy based on converged $\mathbf{w}_e$ will be robust to the adaptive mixed attack schemes. This game process can be formulated as:

$$\begin{aligned} &\min_{w_{ei} \in \mathbf{w}_e} \max_{w_{ai} \in \mathbf{w}_a} g(\mathbf{w}_e, \mathbf{w}_a) \\ &\text{s.t. } g(\mathbf{w}_e, \mathbf{w}_a) = \textstyle\sum_{i=1}^{n} w_{ai} \mathcal{L}\left(a_i(x), y\right) \end{aligned} \tag{5}$$

where $g = ASR$ is the objective function that represents what we want to minimize/maximize, $\mathcal{L}$ is the loss function of the ensemble detector. For $i \in [1, n]$, a_i is a series of base attacks and n is the number of base attack types.

5.2 Solving Approximate Nash Equilibria

In order to obtain the most robust ensemble strategy w_e^*, we need to find the Nash equilibrium solution of the above Minimax Game. Inspired by [3], we convert the Minimax Game into two Markov decision processes, aiming to simplify computation. The two Markov decision are the process of finding optimal w_e^* and the process of finding optimal w_a^*, respectively. Then the Nash equilibrium solution can be represented as (w_e^*, w_a^*).

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_2

With the goal of simplifying the solution process of (w_e^*, w_a^*), we consider solving an approximate Nash equilibrium rather than an exact Nash equilibrium. Since the Bayesian optimization algorithm can save time overhead by referring to previous evaluations when trying the next set of hyper-parameters, it can get better results with a small number of attempts. To this end, we use the Bayesian optimization (BayesOpt) to solve the approximate Nash equilibrium (w_e^*, w_a^*). Details are shown in Algorithm 1.

Algorithm 1 Solving Approximate Nash Equilibria

Input: f, X_a, X_e, S, w_e, w_a
Output: ${w_e}^*$,${w_a}^*$

Initialize f, X_a, X_e, S, w_e, w_a
$w_a = f_s(f, X_a)$
$w_e = f_s(f, X_e)$
while : each $k = 1, 2, 3, ...$ **do**
    while : each $i = 1, 2, 3, ...$ **do**
        $p(y_a \mid D_a, w_a) = fit(M, D_a)$
        $w_{ai} = \arg\max_{w_{ai} \epsilon X_a} S(w_a, p(y|w_a, D))$
        Calculate objective function $y_i = f(w_{ai})$
        $D_a = D_a \bigcup (w_{ai}, y_{ai})$
        Increment i
    end while
    Adversary update attack strategies w_{ai} according to $M(w_a)$
    while : each $j = 1, 2, 3, ...$ **do**
        $p(y \mid D_e, w_e) = fit(M, D_e)$
        $w_{ej} = \arg\min_{w_{ej} \epsilon X_e} S(w_e, p(y|w_e, D_e))$
        Calculate objective function $y_{ej} = f(x_{ej})$
        $D_e = D_e \bigcup (w_{ej}, y_{ej})$
        Increment j
    end while
    Detector update ensemble strategies w_{ej} according to $M(w_e)$
end while

Specifically, there are two loops of training. The first loop is the entire game model, and the second loop is BayesOpt. In the first loops, the adversary and detector sample attack mixed strategy w_a and detector ensemble strategies w_e during each epsilon $c \in \{1, ..., k\}$. The current objective function g is solved and $\mathbf{w}_a^c$ is updated to $\mathbf{w}_a^{c+1}$. Simultaneously, the detector ensemble strategy $\mathbf{w}_e^c$ is also updated by g to obtain $\mathbf{w}_e^{c+1}$. In the second loop, the adversary first randomly based on sample function f_s selects several sets $w_{a1}, w_{a2}, ...w_{an}$ to train the prior function M that is fitted to datasets D_a, D_e, D_t, and get the target values y. Then M is used to fit w_a, y, and acquisition function S is used to select the best w_a^*, and get the new y. Similarly, the detector's parameters are updated in the same way as the adversary. Finally, at a Nash equilibrium, both the adversary and detector will not want to change their strategies (w_a^*, w_e^*) since they will not lead to further benefit.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_2

6 Experiments and Results

In this section, we conduct comprehensive experiments to evaluate the performance of our proposed *NashAE*. The experimental setup and results analysis will be described in detail.

6.1 Experimental settings

· **Datasets:** We use CTU13 [4] and Datacon2020-EMT [1] as encryption malware traffic datasets to evaluate our *NashAE*. Traffic in the two datasets is divided into training set, validation set, and test set in a ratio of 6:2:2.

· **Base Models for Ensemble Malware Detector:** We implement 4 classic machine-learning models as base models, which can be integrated with certain strategies to construct ensemble malware detectors. They are logistic regression (LR) model, support vector machines (SVM) model, decision trees (DT) model, and multi-layer perception (MLP) model, respectively.

· **Baselines of Ensemble Strategy:** To evaluate the robustness performance against adversarial attacks of our *NashAE*, we implement 4 typical ensemble strategies and 2 advanced robust defense strategy as baselines. The typical ensemble strategies include: **1)** Bagging, **2)** Adaboost, **3)** Gradient boosting decision trees (GBDT), and **4)** Stacking. The advanced ensemble strategies include **5)** Omni [14], and **6)** NashRL [3]. Besides, aiming to demonstrate the effectiveness of *NashAE*'s dynamic ensemble structure, we extra implement **7)** BayesAE, which removes adaptive adversaries from the Minimax Game.

· **Evaluation Metrics:** We use three typical machine learning metrics to evaluate the detection performance of malware detectors. To evaluate the attack performance of adversarial attacks, we extra design attack success rate (ASR) metric. To evaluate the robustness against adversarial attacks of malware detectors, we extra design detection rate (DR) metric. Details are as follows:

1. **Typical metrics**: It includes precision (P), recall (R), and F1 score (F1).
2. **ASR**: The attack success rate measures the proportion of traffic with successful attacks in all generated traffic.
3. **DR**: The detection rate represents the proportion of the generated adversarial traffic that is detected by the detector within all adversarial traffic.

· **Experiment preparation:** For the convenience of subsequent experiments, we pre-trained a series of malware detectors on CTU13 dataset and Datacon-EMT dataset, respectively. As shown in Table 1, all of these detectors achieve satisfactory detection performance with high precision, recall, and F1 score.

6.2 Effectiveness Evaluation of Transfer-based Adversarial Attacks

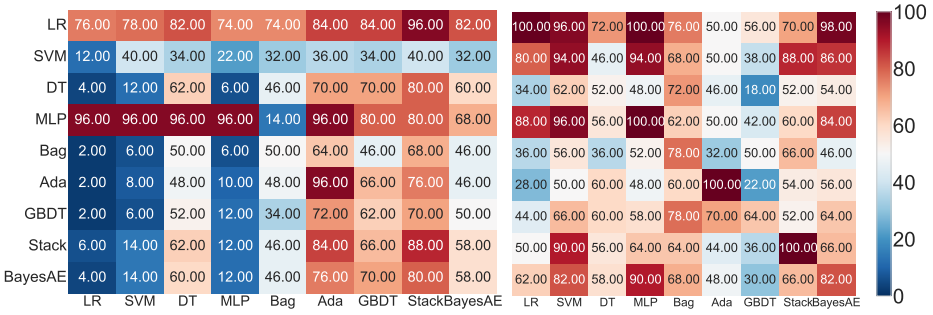
To evaluate the effectiveness of transfer-based adversarial attacks, we perform adversarial attacks on each of the above 11 malware detectors. All the optimal adversarial perturbations are found by NES algorithm. With the goal of transferability evaluation, we add adversarial perturbations generated from the current

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_2

Table 1: Detection performance without adversarial attacks

Detectors	R	P	F1	R	P	F1
	CTU13			Datacon-EMT		
LR	0.8961	0.9363	0.9380	**0.8589**	0.7248	0.7862
SVM	0.9151	0.9845	0.9485	**0.8643**	0.7208	0.7861
DT	**0.9958**	**0.9983**	**0.9970**	0.8381	0.8335	0.8358
MLP	0.9847	**0.9979**	**0.9912**	0.8326	0.8437	0.8381
Bagging	**0.9962**	**0.9979**	**0.9970**	**0.8873**	**0.8528**	**0.8697**
Adaboost	**0.9907**	**0.9953**	**0.9930**	0.8381	0.8464	0.8422
GBDT	**0.9958**	**0.9991**	**0.9975**	**0.9201**	**0.8599**	**0.8890**
Stacking	**0.9958**	**0.9983**	**0.9970**	**0.9136**	0.8486	**0.8799**
Omni	0.9571	**0.9969**	0.9766	**0.9639**	0.7569	0.8479
NashRL	0.9745	0.9899	**0.9901**	**0.9136**	0.8486	**0.8799**
BayesAE	**0.9943**	**0.9979**	**0.9912**	**0.8973**	**0.8638**	**0.8871**
NashAE	**0.9907**	**0.9953**	**0.9930**	**0.8873**	**0.8528**	**0.8697**

detector to the input traffic of the other 10 detectors. Based on this, we can obtain a cross-technology transferability matrix. As shown in Fig. 2, the number on each cell of the matrix represents the ASR based on the transfer between the row and column corresponding detectors. More intuitively, the redder the cell color, the higher the ASR of the corresponding transfer attack. Similarly, the bluer the color, the lower the ASR.

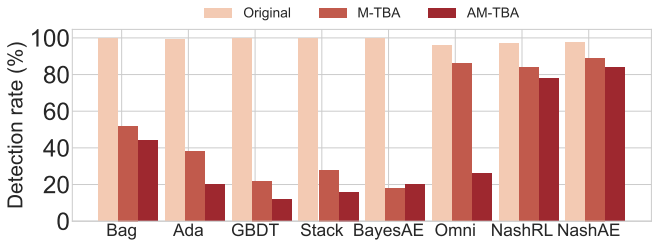


(a) CTU13 dataset (b) Datacon-EMT dataset

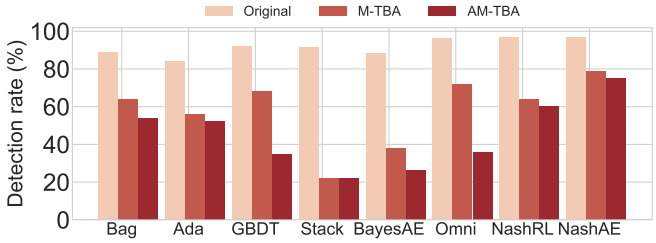
Fig. 2: Transfer-based attack success rates on two datasets.

Specifically, we can find that dark red cells are usually transferred between the same or similar detector models. That is, the more similar the models are, the better the transfer will be. But there is a special case that the adversarial traffic generated for DT are almost useless for LR, which only achieve a 4% attack success rate on the CTU13 dataset. In addition, almost all ensemble models (Bagging, Adaboost, GBDT, Stack) have been attacked successfully with ASR up to 90%. Even though our BayesAE is relatively robust compared with other ensemble methods, it is still vulnerable to specific adversarial attacks. This means that a simple fixed ensemble strategy cannot achieve satisfactory robustness against even black-box adversarial attacks.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_2



(a) CTU13 dataset



(b) Datacon-EMT dataset

Fig. 3: Detection rate of 8 ensemble detectors under two threat models.

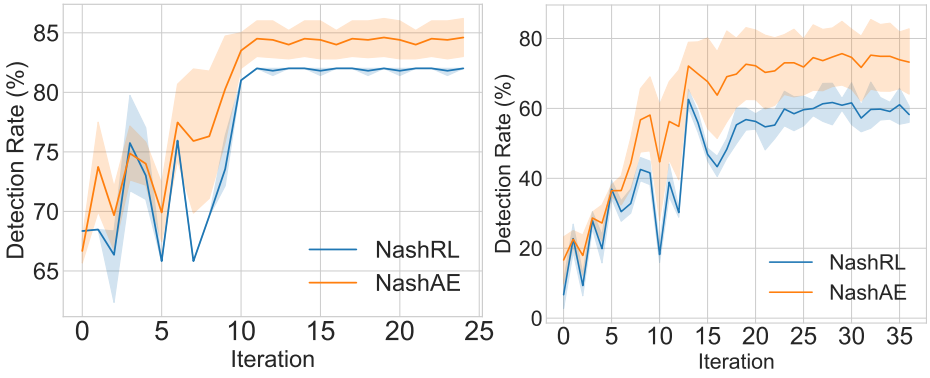
6.3 Robustness Evaluation against M-TBA and AM-TBA

To further evaluate the robustness of ensemble detectors under realistic attack assumptions, we conduct adversarial attacks to five basic ensembles and three advanced ensembles under M-TBA and stronger AM-TBA attacks, respectively. As shown in Fig. 3, compared with the basic detectors, all of the three advanced ensemble detectors (Omni, NashRL and *NashAE*) exhibit strong robustness against an M-TBA attack. For example, on the CTU13 dataset, their detection rate is over 80%, while almost all of other ensemble detectors can only achieve a detection rate below 50%. Under AM-TBA, however, the robustness of Omni is greatly diminished, reaching a *DR* of only 22% on the CTU13 dataset, while NashRL and *NashAE* are significantly better and still achieve around 80% *DR*. Datacon-EMT performs similarly to CTU-13, and our *NashAE* can still attain a detection rate of over 75% despite adaptive attacks.

6.4 Convergence Performance Evaluation

As shown in Fig. 4, we further study the convergence performance of NashRL and *NashAE*. Although their iteration times of converge almost identical on both CTU13 and Datacon-EMT datasets, the detection rate of *NashAE* is steadily better than that of NashRL. Specifically, *DR* of NashRL is only 2.5% worse than NashAE on the CTU13 dataset. However, on the Datacon-EMT dataset, NashRL becomes extremely vulnerable to adversarial attacks after attack power has been doubled. The *DR* metric of NashRL is even less than 60%. In contrast, our NashAE can still achieve a maximum of 75% *DR*. This means that our method can converge to a stable equilibrium solution, and outperform NashRL by over 15% regardless of the attack strength.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_2



(a) CTU13 dataset (b) Datacon-EMT dataset

Fig. 4: Iterations numbers of NashAE and NashRL on two datasets.

7 Conclusion

In this paper, we propose *NashAE* as an ensemble malware detector with dynamic ensemble strategies. As the first step towards training robust malware detectors by adversarial ensemble framework, *NashAE* provides security operation and maintenance personnel with a solution that can deal with powerful adversaries. Our work can be further improved, i.e., *NashAE* uses restricted feature-space attacks to mimic traffic-space attacks, while the traffic should be directly modified in a more realistic traffic-space adversarial attack. To solve this problem, the inverse feature-mapping [11] is a critical technique. However, converting a feature vector into a traffic-space object is difficult due to the feature mapping function is neither invertible nor differentiable. In the future, we will continue to explore solutions to this challenge for a more robust detector.

Acknowledgement. This study was supported in part by the Key Research Program of the Chinese Academy of Sciences under Grant NO. KGFZD-145-23-03, and the State Grid Henan Company's Industrial Unit Science and Technology Project under Grant 2024-KJ-01 (Research on High Security Access Technology for 5G New Distribution Grids).

References

1. https://datacon.qianxin.com/opendata/maliciousstream
2. https://github.com/trusted-ai/adversarial-robustness-toolbox
3. Dou, Y., Ma, G., Yu, P.S., Xie, S.: Robust spammer detection by nash reinforcement learning. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 924–933 (2020)
4. Garcia, S., Grill, M., Stiborek, J., Zunino, A.: An empirical comparison of botnet detection methods. computers & security **45**, 100–123 (2014)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_2

5. Huang, Z., Zhang, T.: Black-box adversarial attack with transferable model-based embedding. arXiv preprint arXiv:1911.07140 (2019)
6. Ilyas, A., Engstrom, L., Athalye, A., Lin, J.: Black-box adversarial attacks with limited queries and information. In: International Conference on Machine Learning. pp. 2137–2146. PMLR (2018)
7. Kariyappa, S., Qureshi, M.K.: Improving adversarial robustness of ensembles with diversity training. arXiv preprint arXiv:1901.09981 (2019)
8. Li, D., Li, Q.: Adversarial deep ensemble: Evasion attacks and defenses for malware detection. IEEE TIFS **15**, 3886–3900 (2020)
9. Li, Y., Song, Y., Jia, L., Gao, S., Li, Q., Qiu, M.: Intelligent fault diagnosis by fusing domain adversarial training and maximum mean discrepancy via ensemble learning. IEEE Transactions on Industrial Informatics **17**(4), 2833–2841 (2020)
10. Novo, C., Morla, R.: Flow-based detection and proxy-based evasion of encrypted malware c2 traffic. In: Proceedings of the 13th ACM Workshop on Artificial Intelligence and Security. pp. 83–91 (2020)
11. Pierazzi, F., Pendlebury, F., Cortellazzi, J., Cavallaro, L.: Intriguing properties of adversarial ml attacks in the problem space. In: 2020 IEEE Symposium on Security and Privacy (SP'20). pp. 1332–1349. IEEE (2020)
12. Qiu, H., Dong, T., Zhang, T., et al.: Adversarial attacks against network intrusion detection in iot systems. IEEE Internet of Things Journal (2020)
13. Rigaki, M.: Adversarial deep learning against intrusion detection classifiers (2017)
14. Shu, R., Xia, T., Williams, L., Menzies, T.: Omni: automated ensemble with unexpected models against adversarial evasion attack. Empirical Software Engineering **27**(1), 1–32 (2022)
15. Tayyab, U.e.H., Khan, F.B., Durad, M.H., Khan, A., Lee, Y.S.: A survey of the recent trends in deep learning based malware detection. Journal of Cybersecurity and Privacy **2**(4), 800–829 (2022)
16. Tramèr, F., Kurakin, A., Papernot, N., et al.: Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204 (2017)
17. Urooj, B., Shah, M.A., Maple, C., Abbasi, M.K., Riasat, S.: Malware detection: a framework for reverse engineered android applications through machine learning algorithms. IEEE Access **10**, 89031–89050 (2022)
18. Wang, H., Wang, Y.: Self-ensemble adversarial training for improved robustness. In: International Conference on Learning Representations (2021)
19. Wu, D., Fang, B., Wang, J., Liu, Q., Cui, X.: Evading machine learning botnet detection models via deep reinforcement learning. In: ICC 2019-2019 IEEE International Conference on Communications (ICC). pp. 1–6. IEEE (2019)
20. Xie, C., Zhang, Z., Zhou, Y., et al.: Improving transferability of adversarial examples with input diversity. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2730–2739 (2019)
21. Xing, X., Jin, X., Elahi, H., Jiang, H., Wang, G.: A malware detection approach using autoencoder in deep learning. IEEE Access **10**, 25696–25706 (2022)
22. Zhang, F., Wang, Y., Liu, S., Wang, H.: Decision-based evasion attacks on tree ensemble classifiers. World Wide Web **23**(5), 2957–2977 (2020)
23. Zhang, F., Wang, Y., Wang, H.: Gradient correlation: Are ensemble classifiers more robust against evasion attacks in practical settings? In: International Conference on Web Information Systems Engineering. pp. 96–110. Springer (2018)
24. Zhou, M., Wu, J., Liu, Y., Liu, S., Zhu, C.: Dast: Data-free substitute training for adversarial attacks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 234–243 (2020)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97632-2_2