# Advancing MobileNet Security: Weighted Adversarial Learning in Convolutional Neural Networks


1st Hakeem Quadri
*Department of Information Technology*
*Victoria University*
Melbourne, Australia
kkeem87200@yahoo.com

2nd Bruce Gu
*Key Laboratory of Computing Power Network and Information Security*
*Qilu University of Technology*
Jinan, China
bruce.gu@vu.edu.au

3rd Hua Wang
*Institute for sustainable Industries and Liveable Cities*
*Victoria University*
Melbourne, Australia
hua.wang@vu.edu.au

4th Sardar Islam
*Institute for sustainable Industries and Liveable Cities*
*Victoria University*
Melbourne, Australia
Sardar.Islam@vu.edu.au



***Abstract*—Convolutional Neural Networks (CNN), particularly low latency models such as MobileNets have excelled in many applications, including object detection in images, speech recognition and natural language processing, but they are vulnerable to subtle perturbations that are virtually imperceptible to the human eye yet can deceive the network into misclassifying images. To enhance the robustness of such CNNs against adversaries, conventional adversarial training methods treat all data points as equally important and susceptible to attack.**

**A weighted adversarial learning algorithm is developed in a Stackelberg game framework. This approach prioritizes data points that are more susceptible to attacks during network training. To further optimize the algorithm, we employ a Reinforcement Learning (RL) agent to fine-tune the hyperparameters of the model, thereby increasing its robustness. Our findings indicate an increased robustness of 66.18% of the Weighted Adversary Reinforced Stackelberg Learning (WARS) against the traditional adversarial training of 64.72% in a one epoch training, using the CIFAR-10 dataset. We conclude that the WARS represents a valuable adversarial training method for bolstering the robustness of low latency CNN models.**



***Index Terms*—convolution neural networks, game theory, Stackelberg games, adversarial training**


## I. Introduction

Pre-trained Convolutional Neural Networks (CNNs) classifiers exhibit high accuracy on natural datasets but are vulnerable to adversarial attacks. These attacks induce misclassifications by introducing noise to the natural datasets. The perturbations added to the datasets are imperceptible to the human eye yet sufficient to deceive the classifier, effectively compromising their reliability in real-world applications such as on medical devices, smartphones and other edge devices where data can be easily manipulated [1].

MobileNets, characterized by compact and shallow architectures compared to larger CNNs exhibit reduced capacity to capture intricate details rendering them more susceptible to adversarial attacks. The increasing importance and complexity of mobile networks make MobileNets appealing targets for cyber adversaries. Adversarial attacks pose a significant threat, particularly on CNN embedded vision applications and mobile devices. Attackers manipulate visual data subtly, introducing perturbations that can lead to incorrect classifications or compromise of the MobileNets [2], [3].

Adversarial Training has emerged as a promising approach to fortify CNNs against these adversarial attacks [4]–[7]. It involves training the models on perturbed data to enhance their robustness [8]–[11]. Standard Adversarial Training (AT) method involves computing an unweighted average of losses across all training data points, treating all adversarial examples equally during training [12]–[16]. However, this approach assumes that adversaries lack incentives or preferences to selectively target specific data points. In reality, attackers may strategically focus on data points that could lead to catastrophic outcomes, even if the classifier exhibited high overall accuracy. Such targeted attacks highlight the logical significance of considering the cost-effectiveness and impact of attacks, urging the incorporation of importance weights during training for better defence.

The sequential interaction between defenders and adversaries can be framed as a Stackelberg game to optimize the defender's performance [17]. In contrast to simultaneous games for adversarial training, sequential games involve first the leader's commitment to a strategy, then a data distribution transformation by the attacker after observing the leaders strategy and finally the defender's selection of a robust model [3], [18]–[21]. Perturbing natural data during the data transformation stage aims at maximizing classification loss [22]–[26], which highights the significance of implementing a weighted


979-8-3503-8567-0/24/$31.00 ©2024 IEEE

adversarial training algorithm to capture the transformed adversarial sample. In our study, we introduce a reinforced weighted adversarial training, conceptualized as a Stackelberg game, to model the interaction between the defender and an adversary. This approach aims to achieve a robust MobileNet CNN. Our approach prioritizes vulnerable data points during adversarial training to minimize classification loss. The Stackelberg game solution yields an optimal pure strategy model with learning parameters that enhance model generalization across perturbation and distribution attacks. The major contributions of this work is listed as follows:

- We improved the accuracy of a Mobilenet CNN model by incorporating an effective yet simple weighting algorithm to traditional adversarial training methods.
- We optimized the model using a reinforcement learning algorithm that learns optimal hyper tuning parameters to increase the robustness of the model [27]–[29].
- We showed that a Stackelberg equilibrium strategy is beneficial for a learner faced with an adversary in a sequential game interaction.

## II. Literature Review

MobileNets are unique for their efficiency in mobile and edge devices primarily due to their depthwise seperable convolutions, which reduces computation in the first few layers [30]. However, studies have revealed that MobileNets are prone to adversarial attacks that can significantly impair their performance in image classification tasks. Even slight perturbations on images can cause substantial declines in classification accuracy [31]–[34]. To counter these vulnerabilities, adversarial training methods have been proposed, aiming to bolster the resilience of deep neural networks against such attacks. Adversarial training methods were initially proposed to enhance the resilience of deep neural networks against adversarial attacks. Over time, this approach has proven to be highly adaptable, finding applications in various domains of machine learning. The core idea revolves around the generation of adversarial examples during the training process, which forces the model to adjust and refine its decision boundaries. Prominent methodologies utilised include the Fast Gradient Sign Method [35] , Projected Gradient Descent (PGD) [36] and adversarial training employing generative models [37].

Research indicates that models trained with single-step adversarial training methods may overfit, reducing their effectiveness against adversaries. However, integrating dropout scheduling into single-step adversarial training can result in more robust models. A hyperparameter introduced to control overfitting enables these models to defend not only against single-step but also multi-step attacks [38]. For instance, Feature-Level Adversarial Training (FLAT) is designed to ensure consistent predictions for both original and adversarial example pairs, and utilizing variational word masks further guides the model to focus on datapoints that enhances accuracy and robustness against adversarial attacks [39]–[42].

Numerous studies have also modelled adversarial training as a simultaneous game between a classifier and an adversary. In such games, the adversary perturbs data using point-wise perturbations to transform the training data, with the goal of increasing misclassification errors for the classifier while avoiding detection [43]–[46]. The problem is formulated as a worst-case min-max game, where both the classifier and the adversary aim to minimize the adversarial loss. Strong perturbation attacks are achieved through Projected Gradient Descent (PGD) to train robust learning models in a single-step min-max interaction [47]. Additionally, results from PGD-based attacks can be emulated using Fast Gradient Sign Method (FGSM) by reducing the curvature along the perturbed direction projected by FGSM. This is accomplished by regularizing the curvature of the attack and restraining the projection to align with those generated by PGD attacks. An introduced hyperparameter controls the curvature along the attack direction and regularizes the model [48]. A game theory framework proposed by Ambar et al. explores attacks and defenses, leading to equilibrium in a simultaneous game setting [2], [8], [43], [49]–[51].

In the context of adversarial attacks on reinforcement learning algorithms, these attacks are presented as generated noises that result in the misclassification of the learning algorithm [48], [52]–[54]. Rajeswaran et al. investigate an ensemble of models for robust reinforcement learning, combining deep neural networks with reinforcement learning to create a robust agent. The interaction between the adversary and the reinforcement learning agent is akin to a min-max game theory formulation [55]. Adversarial training in reinforcement learning enhances robustness against attacks that mislead the reinforcement learning agent into believing it is in a worst-performing trajectory state, leading to sub-optimal actions [56]–[59]. While adversarial training based on a min-max formulation is often overly pessimistic and may not generalize well over test distributions, a more practical approach involves sequential interactions between classifiers and adversaries. In this scenario, the defender initially selects a model while knowing the existence of an optimal adversary. The adversary then chooses a strategy while considering the defender's choice [17]. This hierarchical nature of Stackelberg games provides the defender with a first-mover advantage, constraining the adversary's choices to optimize their own payoff. For example, a game can be modelled as an optimization problem between a data generator and a learner within a Stackelberg game framework [60], [61]. Gao et al. demonstrated the existence of Stackelberg equilibrium that converge to an optimal robust classifier in interactions between Deep Neural Networks (DNNs) and adversaries. Adversaries not only focus on perturbing data but can also manipulate the dataset distribution to maximize classification errors during test time. Traditional adversarial defense mechanisms train models on a uniform training data distribution, which may not generalize well to unseen adversarial data distributions at test time. The Adversarial Risk Importance method is effective in generalizing well under both uniform and non-uniform attacks [62]. Furthermore, Distributionally Robust Optimization (DRO) has been combined with adversarial training to

produce more robust models [63]–[67]. The goal of adversarial training is to reduce classification loss during test time, which necessitates a hierarchical interaction occurring sequentially between classifiers and adversaries.

Combining both Stackelberg game and weighted adversarial learning methods provides an effective defense mechanism that generalizes well across test distributions for a defender. While several works have independently explored game theory frameworks, reinforcement learning and distribution-based robust optimization, this paper introduces a novel approach by combining both Stackelberg games and reinforced weighted adversarial training. The objective is to obtain a classifier that effectively generalizes to both perturbation and targeted attacks particularly those deployed against mobile and edge devices.

## III. System Modelling and Analysis

### A. Preliminaries

We have a training set of $n$ pairs $(x_i, y_i)_{i=1}^{N} \in \mathcal{X} \times \mathcal{Y}$ drawn independently and identically (iid) from a distribution $\mathcal{D}$. Here $x_i$ represents the CIFAR-10 data examples and $y_i$ denotes the corresponding labels. Our primary goal is to develop a robust MobileNet classifier model parameterized by $\theta$ that effectively maps the input space to the output space, denoted as $f_\theta$: $\mathcal{X} \longrightarrow \mathcal{Y}$ while minimizing a loss function $l(x, y;\ \theta)$ on adversarial data $x^{'}$. In this context, we introduce the $L_\infty$ norm metric $d(x, x^{'})$ on $\mathcal{X}$ and a boundary ball $B_\epsilon(x) = \{x^{'} : d(x, x_i) \leq \epsilon\}$ around $x$, an adversary's goal is to perturb the data examples $x_i$ to $x_i^{'}$ within a defined budget $\epsilon > 0$ with the aim of maximizing the adversarial loss $l(f_\theta(x_i^{'}), y_i)$ during the training process.

### B. Stackelberg game

Consider a sequential 2-player non-zero sum Stackelberg game $\mathcal{G}$= $(S_L, S_F, u)$ where $S_l$ and $S_f$ are strategy spaces for the classifier leader and adversary follower of game and $u : S_L \times S_F \longrightarrow R$ is the payoff function. The leader has a set of strategies $s_l \in S_L$ and the followers set of strategies is given by $s_f \in S_F$. For a Stackelberg equilibrium there exist a rational best response mapping function $f : S_L \longrightarrow S_F$ such that $u_2(s_l,\ f(s_l)) \geq u_2(s_l,\ s_f)\ \forall s_l\ \in S_L\ ,\ s_f \in S_F$.

The leader makes the first move by selecting a strategy $s_l \in S_L$ to minimize the $u_1$, knowing the existence of a follower. After knowing $s_l$, the follower picks $s_{f2} \in S_F$ to maximize their own payoff $u_2$ where $s_{f2} = f(s_l)$ . Hence, the Stackelberg equilibrium strategies $(s_l^*, s_f^*)$ pair for leader and follower is $s_l^* \in argmin_{s_l \in S_L} u_1(\ s_l, s_{f2})$ and $s_f^* \in argmax_{s_l \in S_L} u_2(\ s_l^*, s_f)$ respectively such that $u_2(s_l^*, f(s_l^*)) \leq u_2(s_l, s_{f2})$. This gives the leader an advantage that imposes a solution favorable for himself while optimizing against the follower's anticipated strategy $s_{f2}$.

**Proposition 3.1** A Stackelberg equilibrium strategy exists with the defender as the leader and adversary the follower if $S_L$ and $S_F$ are compact sets and $U_L$ and $U_F$ are continuous on $S_L \times S_F$.

**Proof**. Since the rational adversarial response strategy $(s_l,\ f(s_l))$ is a subset of the compact set $S_L \times S_F$ we only need to show that set of adversarial responses is closed. If $(s_l^0, s_f^0)$ is the closure of $\Omega_f$ and $(s_l^n, s_f^n)$ are sequence of points converging to $(s_l^0, s_f^0)$ in $\Omega_f$. We show that $\Omega_f$ is closed and $(s_l^0, s_f^0)$, a point on the boundary, is contained in $\Omega_f$. If $(s_l^0, s_f^0) \notin\ \Omega_f$ then $\exists (s_l^0, s_f^*) \in \Omega_f$ such that $U_f(s_l^0, s_f^*) > U_f(s_l^0, s_f^0)$. Let $U_f(s_l^0, s_f^*) - U_f(s_l^0, s_f^0) = \beta$. since $U_F$ is continuous on $S_L \times S_F$ and $(s_l^{n0}, s_f^{n*}) \to (s_l^0, s_f^*)$ then $\exists\ \delta_1 > 0$ such that $|U_f(s_l^{n0}, s_f^{n*}) - U_f(s_l^0, s_f^*)| < \frac{\beta}{3}$. Similarly, as $(s_l^{n0}, s_f^{n*}) \to (s_l^0, s_f^*)\ \exists\ \delta_2 > 0$ such that $|U_f(s_l^{n0}, s_f^{n0}) - U_f(s_l^0, s_f^0)| < \frac{\beta}{3}$ and $|U_f(s_l^{n0}, s_f^{n0}) - U_f(s_l^0, s_f^0)| < \frac{\beta}{3}\ ,\ \forall (s_l, s_f) \in S_L \times S_F$. Therefore, we have

$|U_f(s_l^{n0}, s_f^{n0}) - U_f(s_l^0, s_f^0)| < \frac{\beta}{3} = U_f(s_l^{n0}, s_f^{n0}) < U_f(s_l^0, s_f^0) + \frac{\beta}{3}$

$U_f(s_l^{n0}, s_f^{n0}) < U_f(s_l^0, s_f^*) - \beta + \frac{\beta}{3} = U_f(s_l^{n0}, s_f^{n0}) < U_f(s_l^0, s_f^*) - \frac{2\beta}{3}$

$$= U_f(s_l^{n0}, s_f^{n0}) < U_f(s_l^{n0}, s_f^*) - \frac{\beta}{3}$$

$$= U_f(s_l^{n0}, s_f^{n0}) < U_f(s_l^{n0}, s_f^*)$$

This contradicts the fact that $U_f(s_l^{n0}, s_f^{n0})$ is a sequence in $U_F$, therefore $(s_l^{n0}, s_f^{n0}) \in U_F$ and $U_F$ is closed.

### C. Adversarial Training as a Stackelberg game

Traditional methods of adversarial training [68] aims to solve a minimax problem between a classifier and attacker by minimizing the loss on the input perturbations. The solution converges to an equilibrium such that for a given dataset $S = \{(x_i, y_i)\}_{i=1}^{n}$, the model $f_\theta$ minimizes the expectation of adversarial loss function as shown

$$\min_\theta \frac{1}{n} \sum_{i=1}^{n} \left\{ \max_{x_i^{'} \in B_\epsilon[x_i]} l(f_\theta(x_i^{'}), y_i) \right\} \tag{1}$$

The model adjust its parameters $\theta$ to the adversarial perturbations by treating all generated adversarial samples $x^{'}$ equally when estimating the adversarial loss at test time. The classifier strategy $s_l \in S_L$ is a parameter $\theta$ that gives minimum training loss $l$ on a training set $(x_i, y_i)_{i=1}^{N}$. The strategy minimizes the payoff empirical risk on the dataset, as shown below:

$$s_l = \min_\theta \mathcal{L}(\theta) = \min_\theta \frac{1}{n} \sum_{i=1}^{n} (l(f_\theta(x_i), y_i)) \tag{2}$$

The payoff function $u_F : S_L \times S_F \longrightarrow R$ of the follower is the adversarial loss derived during attack at test time. After observing the classifier $f_\theta$ the adversary chooses a strategy $s_f \in S_f$ that maximally perturbs the original data. To achieve the attack, the optimal strategy $s_f = \{x^{'} : x + \delta\}$ is the best response to $\theta$ and maximizes the loss $\mathcal{L}$ in equation below. The maximum perturbation $\delta$ is derived using projected gradient descent (PGD) algorithm in the $l_\infty$ norm ball. The payoff function $\mathcal{L}$ of the adversary selecting $s_f$ is given as

$$u_F = \mathcal{L}^{'}(\theta) = \sum_{i=1}^{n} \max_{x_i^{'} \in B_\epsilon(x, \delta)} \left( l(f_\theta(x_i^{'}), y_i) \right) \tag{3}$$

$$s.t\ B_\epsilon(x,\delta) = \{\delta : d(x,x^{'}) \leq \epsilon\}$$

The adversary selects a best response $s_f$ that guarantees a high payoff . The solution to (3) obtains a perturbation $\delta$ which also maximizes $\mathcal{L}^{'}(\theta)$.

In other words, the adversary searches for a strategy $s_f$ obtained using (PGD) that maximizes the adversary's payoff while observing the classifier's strategy $s_l$.

On the other hand, the best response for the leader is calculated by considering the adversary's strategy $s_f = \{x^{'} : x+\delta\}$ as a function of the classifier's payoff $\mathcal{L}(\theta)$. The leaders Stackelberg strategy $s_l^*$ is consequently denoted as

$$s_l^* = \min_\theta \mathcal{L}^{'}(x_i^{'}) = \min_\theta \max_{x_i^{'} \in B_\epsilon(x,\delta)} \Big(l(f_\theta(x_i^{'}), y_i)\Big) \quad (4)$$

### D. Defining the Weighing Parameter $\mathbf{c_i}$

Learning the model parameters requires estimating the loss imposed by potential adversaries. The losses which differ from natural data are derived from adversarial samples generated by adversarial perturbations added to the original samples. The derivative of the summation of individual losses from $x_i^{'}$ in a training batch updates the parameter of the model. To maximise the loss in the inner loop, strong $x_i^{'}$, that is adversarial samples that guarantee high losses, are more represented, weaker $x_i^{'}$ are less represented and $x_i^{'}$ that do not misclassify $y_i$ at all are least represented in the adversarial distribution . The loss in fact, guides the model into ultimately learning the parameter of the model to accurately predict the on the adversarial samples. Afterall, the essence of an adversarial attack is to generate the maximum possible loss and adversarial samples do not contribute equally to the overall loss of the distribution $\mathcal{D}'$.

$$\mathcal{L}' = \mathbb{E}_{(X,Y)\sim D, X' \in B_\epsilon[X,\epsilon]} \left(l(f(X'), Y)\right) \quad (5)$$

A priority attacker selects a strategy $s_f \in S_f$ that not only perturbs the data but also ensures a maximum adversarial payoff loss $\mathcal{L}'$. Not all adversarial samples result in incorrect predictions with PGD attack; therefore, a priority attacker modifies the data distribution $\mathcal{D}$ such that the effective adversarial samples that confidently mislead the model $f_\theta$ into generating outputs different from $y$ are more represented.

For the model to be aware of the underlying distribution of strong adversary samples and generalize effectively over benign adversarial data, we introduce a the weighting mechanism that prioritizes adversary data $x_i^{'}$ during training. Stronger adversarial examples, those that result in misclassifications i.e., $f(x_i) = z$ such that $z \neq y_i$ label $y_i$ with a higher margin are assigned greater weight, while the weaker adversarial examples are given lower weight. The strength of an adversarial sample is determined by its classification margin which is the difference between the probabilities of the wrongly predicted label and the correct label. A larger difference indicates a stronger adversarial sample and vice versa. We define the weight $c_i > 0$ as a function of the classification margin $m$ of the adversarial sample hence we have:

$$m(x,y,f) = \max_{z \neq y} P(f(x) = z) - P(f(x) = y) \quad (6)$$

$$\mathcal{L}^{'}(x_i^{'}) = \sum_{i=1}^{n} \max_{x_i^{'} \in B_\epsilon(x,\delta)} c_i l(f_\theta(x_i^{'}), y_i) \quad (7)$$

$$= \sum_{i=1}^{n} \max_{x_i^{'} \in B_\epsilon(x,\delta)} e^{\varphi m(x_i^{'},y,f)}\ l(f_\theta(x_i^{'}), y_i) \quad (8)$$

Give that $c_i = e^{\varphi m(x_i^{'},y,f)}$ and $\varphi > 0$ is a hypertuning parameter

The Stackelberg equilibrium strategy $s_l^*$ for the leader now becomes

$$s_l^* = \min_{\theta \in \mathcal{H}} \max_{x_i^{'} \in B_\epsilon(x,\delta)} e^{\varphi m(x_i^{'},y,f)} l(f_\theta(x_i^{'}), y_i) \quad (9)$$

### E. Weighted Adversarial Reinforced Training

Adversarial training involves the exploration of hyperparameters to achieve an optimized model. Once a hyperparameter configuration is established, it remains unchanged until the completion of the entire training epoch, resulting in the acquisition of a robust model. We propose an alternative approach, wherein instead of adhering to a single hyperparameter throughout all epochs, we dynamically adjust the hyperparameter during training. This adaptation process aims to yield a better-optimized model for the defender by end of the training. In pursuit of hyperparameter optimization, the defender employs the SARSA (State-Action-Reward-State-Action) algorithm. Specifically, the objective is to learn the hyperparameter denoted as $\varphi$ with the intention of enhancing the accuracy of the selected strategy within a single training epoch. Indeed, the retraining is at the cost of additional overall epochs until an optimal accuracy is reached [51], [58], [69]. A Q-value function $Q(s_l^*, \varphi)$ is estimated using a Stackelberg equilibrium strategy-state $s_l^*$ and an action $\varphi$ from a previous $\varphi^{'}$. The defender takes and action $\varphi$ and observes the next strategy state $s_l^{*^{'}}$ and reward $r$. The reward $r$ ensures that the accuracy of current state is higher than the previous one, the Q-value estimate uses the following update rule:

$$Q(s_l^*, \varphi) = Q(s_l^*, \varphi) + \propto (r + \gamma Q(s_l^{*^{'}}, \varphi^{'}) - Q(s_l^*, \varphi)) \quad (10)$$

where $\propto$ and $\gamma$ is the learning rate and discount factor of the reinforcement learning process.

## IV. Experiment

We conducted experiments using the Weighted Adversary Stackelberg (WAS) Training model and fine-tuned its performance with a Reinforcement Learning (RL) algorithm on a pretrained *MobileNet* [70], resulting in the Weighted Adversarial Reinforced Stackelberg (WARS) model. In our experiment, we employed an adversarial attacker to perturb the CIFAR-10 dataset using the PGD attack. We varied the attack's strength by adjusting the parameter *k*. The perturbed dataset was used to assess the accuracy and robustness of the WAS MobileNet.

We evaluated the adversarial robustness of our WARS model on the CIFAR-10 dataset, benchmarking it against traditional adversarial training methods under PGD attacks.

**Algorithm 1.** Weighted Adversarial Reinforced Stackelberg training

**Inputs**: Pre-trained MobileNet $f_\theta$,dataset $\{x_i\}_{i=1}^N$, batch_size, $k$, epsilon, alpha learning rate;
**Output**: Robust MobileNet;
Observe the model and perturb dataset wrt to the model
**for** $k$ do:
$x^{'} = \Pi_{B_\epsilon(x^{'})} x + \alpha sign(\nabla_x f_\theta(x^{'}))$
$s_l^* -$ Solve Weighted Adversarial Stackelberg game to minimize $\mathcal{L}^{'}(x_i{}^{'})$
**for** mini batches in $\{x_i{}^{'}\}_{i=1}^N$,:
**for** $x_i{}^{'}$ in mini batches:
$\mathcal{L}^{'}(x_i{}^{'}) = e^{\varphi m(x_i^{'},y,f)} l(f_\theta(x_i^{'}), y_i)$
$\theta = \theta - \nabla_\theta \mathcal{L}^{'}(x_i{}^{'})$
Initialize $\gamma, \alpha, \lambda$, $f_\theta(x_i^{'})$
**while** reward>0:
Set $\varphi$=$\begin{cases} \varphi + 1 & if\ state = 0 \\ \varphi - 1 & if\ state = 1 \end{cases}$
$\theta = \min_{\theta \in \mathcal{H}} \max_{x_i^{'} \in B_\epsilon(x,\delta)} e^{\varphi m(x_i^{'},y,f)} l(f_\theta(x_i^{'}), y_i)$
evaluate$f_\theta(x_i^{'})$ accuracy
reward = current accuracy-previous accuracy
$Q(s_l^*, \varphi) = Q(s_l^*, \varphi) + \propto (r + \gamma Q(s_l^{*'}, \varphi^{'}) - Q(s_l^*, \varphi))$
**return** reward, $\varphi$, current accuracy

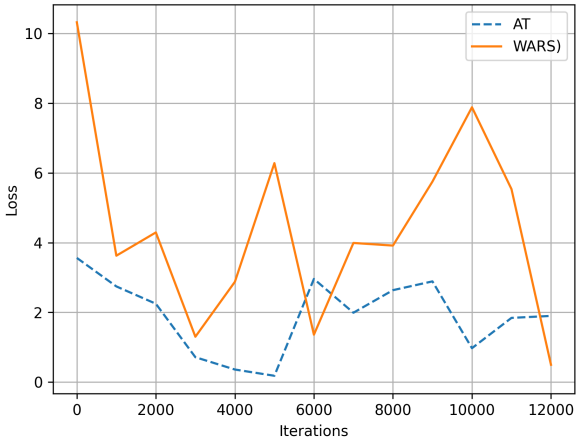


Fig. 1. Epoch training loss for Adversarial Trained and WARS trained mobilenetv2

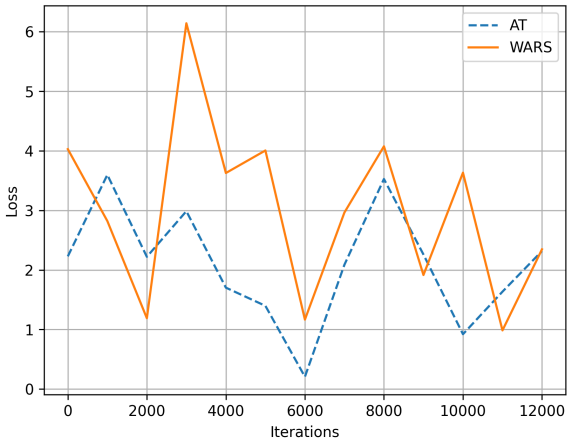


Fig. 2. Epoch training loss for Adversarial Trained and WARS trained shufflenetv2

We applied the WARS algorithm to enhance 3 additional pre-trained models: *ResNet-56*, *shufflenetv2*, and *vgg13_bn* [70] , using different values of $k$, such as 7 and 20 to evaluate the effectiveness of our algorithm. The results demonstrated that our method consistently achieved higher test accuracy compared to traditional adversarial training methods. We used the concept of Natural accuracy $A_n$ representing the accuracy of the pre-trained model $f_\theta$ on the natural CIFAR-10 dataset. After subjecting the model to PGD attacks with varying $k$, denoted as $k$-steps, the corresponding accuracy $A_n^{'}$ of the pre-trained model on the perturbed dataset $x^{'}$ consistently fell below $A_n$, for all the values of k. After training, the resulting WAS model becomes more robust than the initial pre-trained $f_\theta$ showing accuracy $A_R$ consistently greater than $A_n^{'}$ but still less than $A_n$. The WARS model fine-tunes the hyper-parameter $\varphi$ of the WAS to achieve an accuracy $A_R^*$ equals to or greater than $A_R$, such that $A_n^{'} < A_R \leq A_R^*$.

The hyper-parameter $\varphi$ was initially set to 0.7 in the WAS model but improved by the WARS training process for enhanced robustness. As shown in Fig.1, Fig.2, Fig.3 and Fig.4 we observe that in addition to the improved test accuracy, the training loss reduced significantly in a single training epoch, a contrast to traditional adversarial training, which does not exhibit the same behaviour. It's worth noting that the WARS training resulted in a wider range of loss values compared to AT training, and we attribute this to the distribution-aware weight assigned to potential adversarial data points during training, increasing the overall training loss of the model.

We illustrate how an attacker, observing the pre-trained model $f_\theta$, employs PGD to perturb and launch an attack against the target model. The extent of perturbation depends on the selected value of $k$, subsequently reducing the accuracy of the pre-trained models. In our Stackelberg game illustration, the defender selects an equilibrium strategy by observing the attack and choosing a WAS model parameter (through retraining on the perturbed dataset) to minimize losses on the perturbed dataset.

Weighted Adversarial Stackelberg Training leads to improved accuracy compared to the original pre-trained model. Further enhanced learning accuracy is achieved after retraining with a hyper-parameter $\varphi$ . For a moderate preset hyper-parameter $\varphi$ = 7 an overall increase in accuracy is observed across all models. The WARS model further improves the training hyper-parameter during retraining.

As seen in Table 1, a PGD attack with $k$=20 results in stronger attack dataset, significantly reducing the accuracy of all models. Attack steps with $k$=7 used by the attacker also lead to decreased accuracy in the models. Larger $k$ values consistently decrease the overall accuracy of all models. For $k$ values consistently reduce the , the impact of the attack is more pronounced in *ResNet-56*, with accuracy dropping to 10.23% from the initial natural accuracy of 94.46%. The higher the $K$,

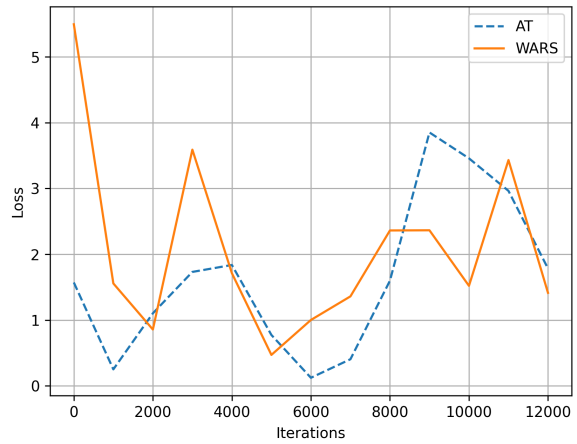


Fig. 3. Epoch training loss for Adversarial Trained and WARS trained RestNet56

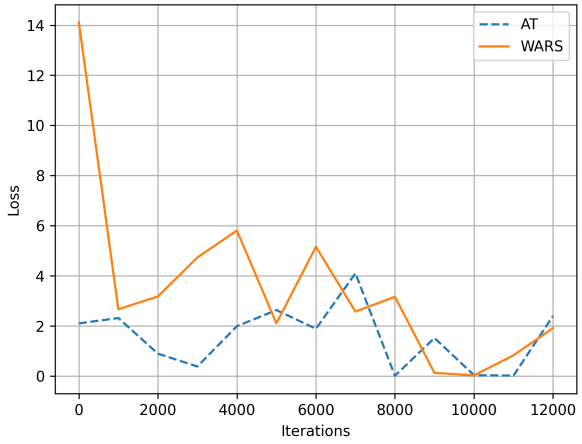


Fig. 4. Epoch training loss for Adversarial Trained and WARS trained vggh

TABLE I
WARS TRAINING FOR VARIOUS PGD STEPS FOR A RESNET-56 MODEL ON CIFAR-10 DATASET.

| **Models** | $A_n\%$ | ***k*** | $A_n^{'}\%$ | $A_R\%$ | $\varphi$ | $A_R^{*\%}$ |
|---|---|---|---|---|---|---|
| vgg13_bn | 94.24 | 20 | 14.17 | 78.22 | 0.8 | 78.22 |
| | | 7 | 17.78 | 78.22 | 0.7 | 78.22 |
| mobilenetv2_x1_4 | 93.88 | 20 | 7.21 | 74.91 | 0.8 | 79.1 |
| | | 7 | 10.66 | 78.11 | 0.9 | 80.01 |
| shufflenetv2_x2_0 | 93.63 | 20 | 12.24 | 78.14 | 0.8 | 79.83 |
| | | 7 | 16.89 | 76.26 | 0.8 | 79.11 |
| ResNet-56 | 94.46 | 20 | 6.81 | 79.83 | 0.8 | 81.23 |
| | | 7 | 10.23 | 79.33 | 1 | 80.45 |

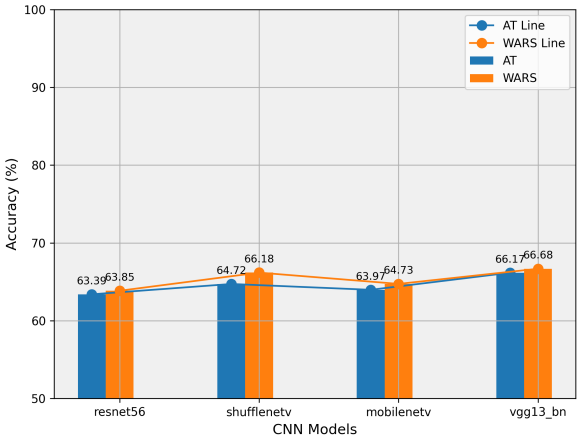


Fig. 5. Accuracy for the different Adversarial Trained and WARS trained CNN in a single Epoch

TABLE II
EPOCH ACCURACY OF THE WAS TRAINING FOR *k*=7 ON VARIOUS CNN MODELS USING CIFAR-10 DATASET.

| **Models** | ***E*=2** | ***E*=3** | ***E*=4** | ***E*=5** | ***E*=6** | $A_R^*\%$ |
|---|---|---|---|---|---|---|
| **vgg13_bn** | **79.46** | 78.29 | 78.68 | 77.32 | 78.42 | 78.22 |
| **mobilenetv2** | 78.56 | 77.73 | 77.96 | 77.57 | **78.68** | 79.12 |
| **Shufflenetv2** | 77.06 | 76.68 | 78.39 | **78.89** | 77.99 | 79.83 |
| **ResNet-56** | 79.52 | 80.18 | 79.99 | 79.92 | **80.19** | 81.23 |

the greater the image distortion, and even when the distortion is imperceptible, the attack still significantly reduces the model's accuracy. After retraining, using distribution-aware Stackelberg training, the accuracy improves to 60.67% , and the WAS model fine-tunes it further to an accuracy of 65.67% with a WARS $\varphi$ of 1.0. The training involved 8 epochs for the WARS training, with additional epochs based on when the model reaches optimal accuracy. For the *ResNet-56* model, default epoch training and adversarial accuracy reached 78.11%, and the WARS trained model optimized the accuracy to 80.01% with a $\varphi$ of 0.8.

From Table II, epoch accuracy for WAS training gradually improves after each epoch from an initially low $A_n^{'}$ of the original model. The original pre-trained model exhibits reduced accuracy after the attack, with *MobileNet* showing an $A_R$ of 10.66%, dropping to 78.11% at the final epoch after achieving 78.68% accuracy. However, the WARS model fine-tunes the model back to an optimized accuracy of 80.01%. The *ResNet-56* model's accuracy is optimized to 80.45% after reaching an $\varphi$ 1.0, up from a previous WAS accuracy of 79.33%, while the *vgg13_bn* accuracy for both WAS and WARS training remained 78.22% at the default $\varphi$ of 0.7.

### A. Discussion

In this research, we have developed a novel adversarial training approach for MobileNet CNNs, conceptualizing it as a dynamic interaction within a WAS game framework.

TABLE III
EPOCH ACCURACY OF THE WAS TRAINING FOR $k$=20 ON VARIOUS CNN MODELS USING CIFAR-10 DATASET.

| **Models** | ***E*=1** | ***E*=2** | ***E*=3** | ***E*=4** | ***E*=5** | $A_R^*\%$ |
|---|---|---|---|---|---|---|
| **vgg13_bn** | 77.33 | 77.53 | 77.75 | 77.79 | 78.22 | 78.22 |
| **mobilenetv2** | 76.73 | 77.23 | 77.98 | 79.35 | 75.92 | 80.01 |
| **Shufflenetv2** | 76.19 | 76.12 | 77.94 | 78.75 | 77.93 | 79.11 |
| **ResNet-56** | 78.86 | 76.58 | 80.2 | 78.52 | 77.73 | 80.45 |

By strategically emphasizing adversarial data points during training, our methodology has substantially improved the model's accuracy. This is achieved by prioritizing adversarial inputs that are more likely to cause misclassifications, thereby training the MobileNet model to develop a bias that enhances its resilience during adversarial attacks.

When comparing our WAS model to traditional AT methods, we observe a notable superiority in terms of robustness under adversarial conditions. Although the WAS model initially shows a broader range of training losses compared to AT models, it demonstrates a more rapid decrease in training loss within a single epoch, particularly when applied to dataset like CIFAR-10, tailored for MobileNet's architecture.

Moreover, our research introduces the WARS training methodology. This refined approach further strengthens the MobileNet model's resilience against adversarial attacks. Our empirical findings, as detailed in the accompanying tables, show consistent enhancements in the performance of MobileNet across various levels of$\varphi$ increments in the training process. This iterative and strategic reinforcement leads to a discernible improvement in accuracy with each successive training epoch, underscoring the efficacy of the WARS approach in crafting a more robust MobileNet CNN.

## V. CONCLUSION

In this paper, we have designed a novel adversarial training methodology, conceptualized as a Weighted Adversarial Stackelberg game, specifically tailored for training a robust MobileNet CNN. Our research demonstrates the effectiveness of the Stackelberg equilibrium model in enhancing MobileNet's resilience against adversarial attacks. We further augment this model's robustness by incorporating a SARSA algorithm, which acts as a defensive mechanism, fine-tuning the MobileNet architecture to counteract such attacks more effectively, we also showed the effectiveness of our methods on other CNN models.

Our approach in the Stackelberg game formulation centres on assigning asymmetric weights that focus more on adversarial data points during testing. This strategy significantly reduces misclassification errors in MobileNet. We derive a pure strategy model with optimized learning parameters by solving the Stackelberg game. This outcome empowers the MobileNet model to generalize more effectively and exhibit increased robustness to targeted and perturbation attacks.

## REFERENCES


[1] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *Proceedings of the IEEE*, 2018.

[2] C. Szegedy, W. Zaremba, and I. Sutskever, "Intriguing properties of neural networks," *ICLR 2014, OpenReview*, 2014.

[3] M. Lechner and A. Amini, "Revisiting the adversarial robustness-accuracy tradeoff in robot learning," 2022.

[4] J. Yin, M. Tang, J. Cao, and H. Wang, "Apply transfer learning to cybersecurity: Predicting exploitability of vulnerabilities by description," *Knowledge-Based Systems*, vol. 210, 10 2020.

[5] Y.-F. Ge, M. Orlowska, J. Cao, H. Wang, and Y. Zhang, "Mdde: multi-tasking distributed differential evolution for privacy-preserving database fragmentation," *The VLDB Journal*, vol. 31, pp. 1–19, 01 2022.

[6] J. Zhang, H. Li, X. Liu, Y. Luo, F. Chen, and H. Wang, "On efficient and robust anonymization for privacy protection on massive streaming categorical information," *IEEE Transactions on Dependable and Secure Computing*, vol. PP, pp. 1–1, 09 2015.

[7] J.-Y. Li, K.-J. Du, Z.-H. Zhan, H. Wang, and J. Zhang, "Distributed differential evolution with adaptive resource allocation," *IEEE transactions on cybernetics*, vol. PP, 03 2022.

[8] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *n: ICLR 2015, OpenReview*, 2015.

[9] J. Yin, M. Tang, J. Cao, M. You, H. Wang, and M. Alazab, "Knowledge-driven cybersecurity intelligence: Software vulnerability co-exploitation behaviour discovery," *IEEE Transactions on Industrial Informatics*, 2022.

[10] J. Shu, X. Jia, K. YANG, and H. Wang, "Privacy-preserving task recommendation services for crowdsourcing," *IEEE Transactions on Services Computing*, vol. PP, pp. 1–1, 01 2018.

[11] Y. Zhang, Y. Shen, H. Wang, J. Yong, and X. Jiang, "On secure wireless communications for iot under eavesdropper collusion," *IEEE Transactions on Automation Science and Engineering*, vol. 13, pp. 1–13, 12 2015.

[12] A. Kurakin, I. Goodfellow, and B. S, "Adversarial examples in thephysical world," *Proceedings of the IRE*, Jul. 2016.

[13] N. Carlini and A. A., "Simple black-box adversarial attacks," *arXiv preprint arXiv*, 2019.

[14] J. Yin, M. Tang, J. Cao, H. Wang, M. You, and Y. Lin, "Vulnerability exploitation time prediction: an integrated framework for dynamic imbalanced learning," *World Wide Web*, pp. 401–423, 01 2022.

[15] H. Wang, J. Cao, and Y. Zhang, "A flexible payment scheme and its role-based access control," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, pp. 425– 436, 04 2005.

[16] H. Wang, Y. Zhang, and J. Cao, "Effective collaboration with information sharing in virtual universities," *IEEE Trans. Knowl. Data Eng.*, vol. 21, pp. 840–853, 06 2009.

[17] L. Dritsoula and P. Loiseau, "A game-theoretic analysis of adversarial classification," *IEEE Transactions on Information Forensics and Security*, 2017.

[18] J. Bose and G. Gidel, "Adversarial example games," *Proc NeurIPS*, 2020.

[19] K. Cheng, L. Wang, Y. Shen, H. Wang, Y. Wang, X. Jiang, and H. Zhong, "Secure k-nn query on encrypted cloud data with multiple keys," *IEEE Transactions on Big Data*, vol. PP, pp. 1–1, 05 2017.

[20] Y.-F. Ge, H. Wang, E. Bertino, Z.-H. Zhan, J. Cao, Y. Zhang, and J. Zhang, "Evolutionary dynamic database partitioning optimization for privacy and utility," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–17, 2023.

[21] H. Wang, Y. Zhang, J. Cao, and V. Varadharajan, "Achieving secure and flexible m-services through tickets," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 33, pp. 697 – 708, 12 2003.

[22] K. Grosse, D. Pfaff, and M. Smith, "The limitations of model uncertainty in adversarial settings," 2018.

[23] H. Zhang and Y. Yu, "Theoretically principled trade-off between robustness and accuracy," *Vehicular Technology, IEEE Transactions on*, 2019b.

[24] S. Siuly, Alçin, H. Wang, Y. Li, and P. Wen, "Exploring rhythms and channels-based eeg biomarkers for early detection of alzheimer's disease," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. PP, pp. 1–15, 04 2024.

[25] M. N. A. Tawhid, S. Siuly, K. Wang, and H. Wang, "Automatic and efficient framework for identifying multiple neurological disorders from eeg signals," *IEEE Transactions on Technology and Society*, vol. PP, pp. 1–1, 03 2023.

[26] C. Wang, B. Sun, K.-J. Du, J.-Y. Li, Z.-H. Zhan, S.-W. Jeon, H. Wang, and J. Zhang, "A novel evolutionary algorithm with column and sub-

block local search for sudoku puzzles," *IEEE Transactions on Games*, vol. PP, pp. 1–11, 01 2023.

[27] T. Huang, Y.-J. Gong, W.-n. Chen, H. Wang, and J. Zhang, "A probabilistic niching evolutionary computation framework based on binary space partitioning," *IEEE Transactions on Cybernetics*, vol. PP, pp. 1–14, 03 2020.

[28] W.-L. Liu, Y.-J. Gong, W.-n. Chen, Z. Liu, H. Wang, and J. Zhang, "Coordinated charging scheduling of electric vehicles: A mixed-variable differential evolution approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, pp. 1–16, 10 2019.

[29] Z.-G. Chen, Z.-H. Zhan, H. Wang, and J. Zhang, "Distributed individuals for multiple peaks: A novel differential evolution for multimodal optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. PP, pp. 1–1, 10 2019.

[30] A. G. Howard, Z. Menglong, C. Bo, K. Dmitry, W. Weijun, W. Tobias, A. Marco, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017.

[31] P. Hai, L. Zechun, H. Dang, S. Marios, C. Kwang-Ting, and Z. Shen, "Binarizing mobilenet via evolution-based searching," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020,*, pp. pp. 13 420–13 429, 2020.

[32] C. Yupeng, J. Felix, G. Qing, F. Huazhu, X. Xiaofei, L. Shang-Wei, L. Weisi, and Y. Liu, "Adversarial exposure attack on diabetic retinopathy imagery," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020,*, 2020.

[33] A. Alvi, S. Siuly, and H. Wang, "A long short-term memory based framework for early detection of mild cognitive impairment from eeg signals," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. PP, pp. 1–14, 01 2022.

[34] W. Shi, W.-n. Chen, S. Kwong, J. Zhang, H. Wang, G. Tianlong, H. Yuan, and J. Zhang, "A coevolutionary estimation of distribution algorithm for group insurance portfolio," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. PP, pp. 1–15, 07 2021.

[35] E. Wong and L. Rice, "Fast is better than free: Revisiting adversarial training," 2020.

[36] Y. Wang, X. Ma, and J. Bailey, "On the convergence and robustness of adversarial training," 2021.

[37] K. Murat, S. Christopher, G. Alexandros, and S. Dimakis, Vishwanath, "Causalgan: Learning causal implicit generative models with adversarial training," 2017.

[38] B. Vivek, "Adversarial training with dropout scheduling," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000.

[39] H. Chen, ". adversarial training for improving model robustness?look at both prediction and interpretation," *ICLR*, 2022.

[40] E. Kabir, A. Mahmood, H. Wang, and A. Mustafa, "Microaggregation sorting framework for k-anonymity statistical disclosure control in cloud computing," *IEEE Transactions on Cloud Computing*, vol. PP, pp. 408–417, 08 2020.

[41] H. Wang, Y. Zhang, and J. Cao, "Ubiquitous computing environments and its usage access control," vol. 152, 01 2006, p. 6.

[42] T. Huang, Y.-J. Gong, S. Kwong, H. Wang, and J. Zhang, "A niching memetic algorithm for multi-solution traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 3, pp. 508–522, 2019.

[43] A. Madry, A. Makelov, and L. Schmidt, "Towards deep learning models resistant to adversarial attacks," *nternational Conference on Learning Representations*, 2018.

[44] X. Sun, H. Wang, J. Li, and Y. Zhang, "Injecting purpose and trust into data anonymisation," *Computers  Security*, vol. 30, pp. 332–345, 07 2011.

[45] J. Zhang, X. Tao, and H. Wang, "Outlier detection from large distributed databases," *World Wide Web*, vol. 17, 07 2014.

[46] F. Liu, X. Zhou, J. Cao, Z. Wang, W. Tianben, H. Wang, and Y. Zhang, "Anomaly detection in quasi-periodic time series based on automatic data segmentation and attentional lstm-cnn," *IEEE Transactions on Knowledge and Data Engineering*, vol. PP, pp. 1–1, 08 2020.

[47] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. Ghaoui, and J. MI, "Theoretically principled trade-off between robustness and accuracy," *in International Conference on Machine Learning (ICML)*, 2019.

[48] H. Tianjin, M. Vlado, P. Yulong, and P. Mykola, ". bridging the performance gap between fgsm and pgd adversarial training," *IEEE*, 2020.

[49] Y. Wang, Y. Shen, H. Wang, J. Cao, and X. Jiang, "Mtmr: Ensuring mapreduce computation integrity with merkle tree-based verifications," *IEEE Transactions on Big Data*, vol. 4, no. 3, pp. 418–431, 2016.

[50] Y. Zhang, Y. Shen, H. Wang, Y. Zhang, and X. Jiang, "On secure wireless communications for service oriented computing," *IEEE Transactions on Services Computing*, vol. PP, pp. 1–1, 09 2015.

[51] Y.-F. Ge, E. Bertino, H. Wang, J. Cao, and Y. Zhang, "Distributed cooperative coevolution of data publishing privacy and transparency," *ACM Transactions on Knowledge Discovery from Data*, vol. 18, 08 2023.

[52] Y. Zhang, Y. Gong, Y. Gao, H. Wang, and J. Zhang, "Parameter-free voronoi neighborhood for evolutionary multimodal optimization," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 335–349, 2020.

[53] S. Supriya, S. Siuly, H. Wang, and Y. Zhang, "Eeg sleep stages analysis and classification based on weighed complex network features," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. PP, pp. 1–11, 11 2018.

[54] M. Peng, Q. Xie, H. Wang, Y. Zhang, and G. Tian, "Bayesian sparse topical coding," *IEEE Transactions on Knowledge and Data Engineering*, vol. PP, pp. 1–1, 06 2018.

[55] P. Lerrel, D. James, S. Rahul, and G. Abhinav, "Robust adversarial reinforcement learning," 2017.

[56] P. Anay, T. Zhenyi, L. Shuijing, B. Gautham, and C. Girish, "Robust deep reinforcement learning with adversarial attacks," 2017.

[57] Z.-J. Wang, Z.-H. Zhan, Y. Lin, W.-J. Yu, H. Wang, S. Kwong, and J. Zhang, "Automatic niching differential evolution with contour prediction approach for multimodal optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. PP, pp. 1–1, 04 2019.

[58] M. Peng, J. Zhu, H. Wang, X. Li, Y. Zhang, X. Zhang, and G. Tian, "Mining event-oriented topics in microblog stream with unsupervised multi-view hierarchical embedding," *ACM Transactions on Knowledge Discovery from Data*, vol. 12, pp. 1–26, 04 2018.

[59] M. Peng, W. Gao, H. Wang, Y. Zhang, J. Huang, Q. Xie, G. Hu, and G. Tian, "Parallelization of massive textstream compression based on compressed sensing," *ACM Transactions on Information Systems*, vol. 36, pp. 1–18, 08 2017.

[60] A. S. Chivukula and X. Yang, "Game theoretical adversarial deep learning with variational adversaries," *in IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 8, pp. pp. 3568–3581, 2021.

[61] L. Zhang, T. Zhu, F. Khadeer, D. Ye, and W. Zhou, "A game-theoretic method for defending against advanced persistent threats in cyber systems," *IEEE Transactions on Information Forensics and Security*, vol. 18, no. 8, pp. pp.1349–1364, 2023.

[62] H. Zeng, Zhu, and C. Goldstein, "Are adversarial examples created equal? a learnable weighted minimax risk for robustness under non-uniform attacks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, pp. 10 815–10 823, 2021.

[63] M. Staib and S. Jegelka, "Distributionally robust optimization and generalization in kernel methods," *Advances in Neural Information Processing Systems*, vol. 35, no. 12, pp. 10 815–10 823, 2019.

[64] J.-Y. Li, Z.-H. Zhan, H. Wang, and J. Zhang, "Data-driven evolutionary algorithm with perturbation-based ensemble surrogates," *IEEE Transactions on Cybernetics*, vol. PP, pp. 1–13, 08 2020.

[65] E. Kabir and H. Wang, "Conditional purpose based access control model for privacy protection," vol. 92, 01 2009, pp. 137–144.

[66] E. Kabir, "A role-involved purpose-based access control model," *Information Systems Frontiers*, vol. 14, pp. 809–822, 07 2012.

[67] X. Sun, H. Wang, J. Li, and J. Pei, "Publishing anonymous survey rating data," *Data Mining and Knowledge Discovery*, vol. 23, pp. 379–406, 11 2011.

[68] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and V. A, "Towards deep learning models resistant to adversarial attacks," 2018.

[69] J.-Q. Yang, Q.-T. Yang, K.-J. Du, C.-H. Chen, H. Wang, S.-W. Jeon, J. Zhang, and Z.-H. Zhan, "Bi-directional feature fixation-based particle swarm optimization for large-scale feature selection," *IEEE Transactions on Big Data*, vol. PP, pp. 1–14, 01 2022.

[70] Chenyaofo, "https://github.com/chenyaofo/pytorch-cifar-models/tree/master," 2021.