TECHNICAL UNIVERSITY OF MUNICH

Seminar -
Adversarial and Secure Machine Learning

Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models

Author	Moritz Müller
Supervisor	Nicolas Müller (TUM Department of Informatics & Fraunhofer-Institute for Applied and Integrated Security (AISEC))

Sept 2021

Contents

1 Introduction 1

2 Evasion Attacks 1
2.1 Problem Statement 1
2.2 Different Types of Evasion Attacks 2

3 The Boundary Attack 3
3.1 Intuition 3
3.2 Mathematical Foundations 4
3.2.1 Initialization 4
3.2.2 Initial Projection 5
3.2.3 Random Walk 5
3.2.4 Convergence 6
3.3 Dynamical Hyperparameter Adjustment 6

4 Results and Evaluation 7
4.1 Untargeted Attacks 7
4.2 Targeted Attacks 8
4.3 Evaluation of the Boundary Attack 9
4.3.1 Advantages 9
4.3.2 Room for Improvement 9

5 State of the Art 10
5.1 Query Costs 10
5.2 Defense Strategy 11

6 Conclusion 12

Bibliography 13

1 Introduction

Machine Learning (ML) models are vulnerable to small and structured perturbations of their input data, which often are imperceptible to us humans [1]. This means, small input changes can lead to a different classification decision, where humans would not perceive any (significant) changes.

In the context of the presented seminar paper by Brendel et al. [2], we will take a look at the generation of such imperceptible perturbations - the so called *Adversarial Examples* (AE). They can be found through different types of *Evasion Attacks*. In contrast to that, *Poisoning Attacks* - as the second attack variant - manipulate the training dataset before the training, which is why we will not go deeper into these kind of attacks [3]. Attackers can use AEs to trick ML models into predicting any class different from the original or even a specific target class. This becomes a problem, when the model classification is a safety critical feature, like in the field of autonomous driving or facial recognition systems [4]. Therefore, new defense and detection strategies are developed on par with the emerging threat models, to continuously improve the model robustness.

Brendel et al. introduce a new, powerful approach to generate competitive AEs, against which - up to the paper release - no defense strategy exists. By formulating the so called *Boundary Attack* (BA), which only makes use of the final model decision, a much more realistic threat scenario, namely that of hidden Black-Box Models, is addressed [2].

Within the scope of this work, I illustrate the formalization of Evasion Attacks and compare the BA to different approaches in this field (2). After introducing the mathematical foundations and implementation of the BA (3), I give further discussions on its potentials and drawbacks (4). Finally, I present state of the art optimizations and new defense strategies against the BA, to derive possible directions for future work (5).

2 Evasion Attacks

2.1 Problem Statement

As stated in Chapter 1, the goal of an attacker is to generate AEs to trick the defender, i.e. a ML model. The goal of Evasion Attacks can be formalized by a standard definition. Given an input image $\boldsymbol{o}$ of some class x and a perturbation $\boldsymbol{\eta}$, find an adversarial image $\tilde{\boldsymbol{o}}$ of a class, different from x, such that $\tilde{\boldsymbol{o}} = \boldsymbol{o} + \boldsymbol{\eta}$, where the l_p-norm $||\boldsymbol{\eta}||$ should be small - ideally invisible to us humans. We can formulate this as a simple optimization problem, where $d(\tilde{\boldsymbol{o}})$ and $d(\boldsymbol{o})$ refer to the final decision or prediction of the same ML model [1]:

$$min_{\tilde{\boldsymbol{o}}} = ||\tilde{\boldsymbol{o}} - \boldsymbol{o}||$$

$$s.t.\ d(\tilde{\boldsymbol{o}}) \neq d(\boldsymbol{o})$$

The goal is to minimize the distance between $\tilde{\boldsymbol{o}}$ and $\boldsymbol{o}$ - which is nothing else but the perturbation $\boldsymbol{\eta}$ - under some l_p-norm. The constraint ensures, that the attacked model discriminates between the two image classes based on some adversarial criterion. In the Computer Vision (CV) domain this criterion can be one of the two main scenarios. Either, the adversarial image is from any but the original class (untargeted) or a specific adversarial class (targeted).

2.2 Different Types of Evasion Attacks

During the course of this seminar, we got to know many different attacks such as the Fast Gradient Sign Method (FGSM) [5] or the Black-Box Attack strategy described in [6]. All of these attacks can be subcategorized into five main categories, where each category needs less and less information about the attacked ML model.

The first category are **Gradient-Based Attacks** (GBA), which require full access to the attacked model and its gradients [2]. Many of them - like the FGSM [5] - make use of the backpropagation algorithm, to compute gradients on the input image. This runs at $O(p * n)$ time complexity, with p pixels in the input image and n iterations, which are - dependent on the method - typically small. The runtime of the FGSM for example reduces to $O(1)$, since only a single iteration step is taken. A common defense strategy against those methods is gradient masking which implicitly or explicitly adds non-differentiable elements to the model (e.g. defensive distillation or saturated non-linearities) [2].

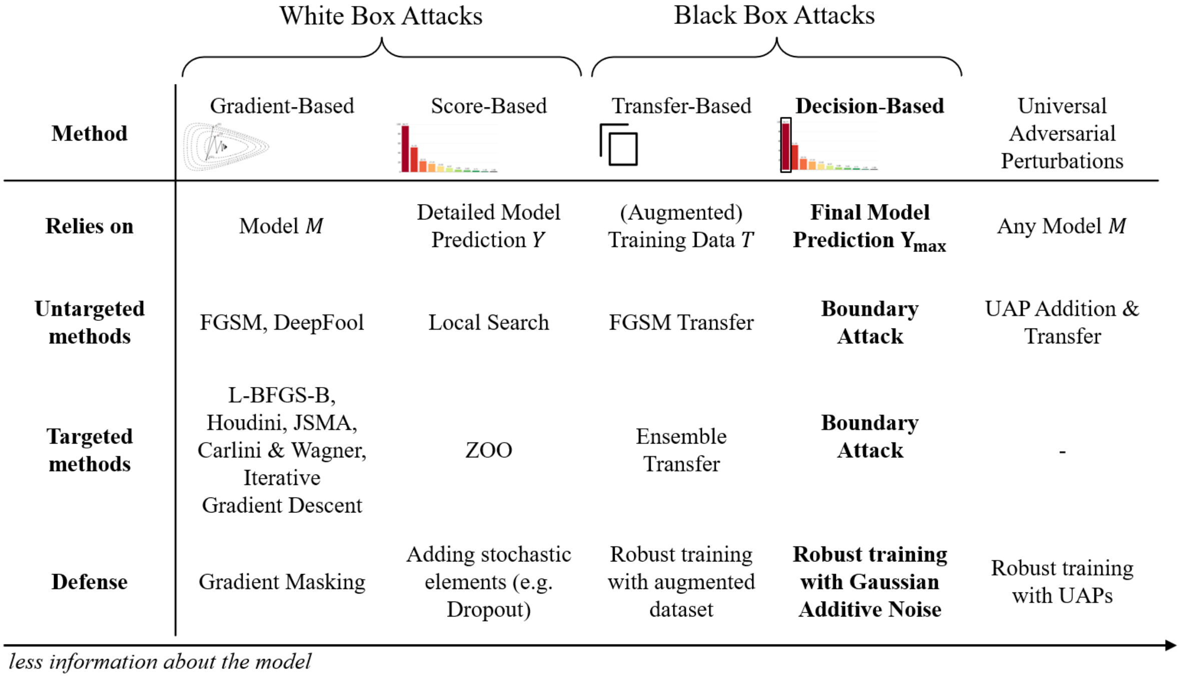
In contrast to that, **Score-Based Attacks** (SBA) only make use of the predicted model scores, to numerically estimate the model gradients [2]. Their space-time complexity highly depends on the specific method and its implementation. State of the art defense strategies against such attacks are adding stochastic elements such as Dropout to the model or robust training to inhibit the possibility for numerical gradient estimations.

The third category are **Transfer-Based Attacks** (TBA), which do not need any information about the model, except for its (synthetic) training dataset [2]. The idea is to train a substitute model - as shown in the Black-Box Attack [6] -, from which AEs can be constructed. Note, that the training of a Deep Neural Network (DNN) takes $O(w * t * e)$ time complexity, with w weights, t training samples and e epochs. Based on the concept of transferability [7], [8], the generated AEs can be used to attack the original model as well. Since we apply GBAs on the substitute model, which often run in constant time, the overall space-time complexity for finding AEs is highly impacted by training the substitute model. To defend against these attacks, we can anticipate the transfer by applying robust training, which augments the original dataset by such AEs.

The fourth category introduced by Brendel et al. are **Decision-Based Attacks** (DBA), which only require access to the final $arg\,max$ prediction of the attacked model. With the BA, the first competitive attack in this category is introduced, which will run in $O(k)$ time and $O(1)$ space, where k is the number of predefined iterations. By the time of the paper release, there exists no efficient defense strategy. Though, I will present a model randomization technique [9] as state of the art defense against BAs in Chapter 5.2.

The final category introduced in the seminar are attacks, which only make use of a **Universal Adversarial Perturbation** (UAP) [8]. Since this perturbation only has to be found once, to turn any image into an AE, but also can be transferred across different ML models, they do not need any information about the attacked model at all. The property of universal transferability also relativizes the inefficiency of the UAP algorithm. Note that nowadays both, efficient algorithms to compute an UAP as well as efficient defense strategies such as Universal Adversarial Training [10] for example, exist.

Note, that a state of the art defense strategy against all Evasion Attacks is robust or adversarial training, which augments the training dataset with AEs identified by a certain attack [3]. Furthermore, there exist limitations for the amount of attacks, a single classifier can defend against when making use of ensemble adversarial training [11].



	White Box Attacks		Black Box Attacks		
Method	Gradient-Based	Score-Based	Transfer-Based	**Decision-Based**	Universal Adversarial Perturbations
Relies on	Model *M*	Detailed Model Prediction *Y*	(Augmented) Training Data *T*	**Final Model Prediction $\mathbf{Y_{max}}$**	Any Model *M*
Untargeted methods	FGSM, DeepFool	Local Search	FGSM Transfer	**Boundary Attack**	UAP Addition & Transfer
Targeted methods	L-BFGS-B, Houdini, JSMA, Carlini & Wagner, Iterative Gradient Descent	ZOO	Ensemble Transfer	**Boundary Attack**	-
Defense	Gradient Masking	Adding stochastic elements (e.g. Dropout)	Robust training with augmented dataset	**Robust training with Gaussian Additive Noise**	Robust training with UAPs

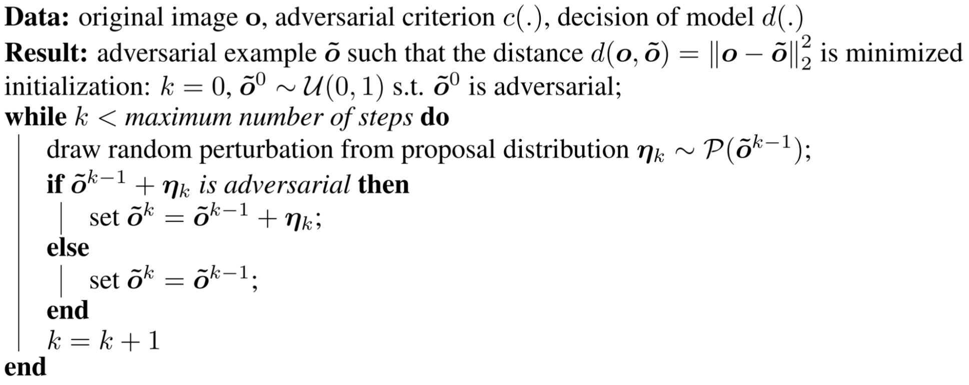
Figure 1: Different Types of Evasion Attacks [2]

3 The Boundary Attack

Brendel et al. introduce Decision-Based Attacks as a new powerful category in the field of Evasion Attacks (see Chapter 2.2). The main problem of other attacks from categories such as GBA or SBA is that they rely on model information, which is often not accessible in a real-world scenario. The idea of the DBA is to categorize attacks, which only require access to the final decision of Black-Box models to generate AEs. This principle addresses a much more realistic scenario for real-world applications (e.g. self-driving cars, home assistants, etc.), where the model is typically hidden on some external server. On the other side a new approach for the generation of AEs has to be formalized. Therefore, the paper proposes the Boundary Attack, which inverts the dynamics of previously seen attacks by starting at some AE and iteratively perturbing it until it becomes as close as possible to the original image.

3.1 Intuition

By heart, the Boundary Attack combines rejection sampling, directed by a proposal distribution with a dynamic Trust Region method to find minimal distance AEs along the decision boundary of a model [2]. The four main stages of the algorithm are its (1) Initialization, (2) Initial Projection, (3) Random Walk and (4) Convergence, which are covered in-depth in Chapter 3.2. The core algorithm as it is introduced by Brendel et al. can be found in Figure 2. Besides the original image $\boldsymbol{o}$ and the model decision $d(.)$, the algorithm takes an adversarial criterion $c(.)$ as input. The idea is to define the criterion, which a newly found AE has to fulfill for a chosen attack. In the CV domain for example, this could either be the untargeted setting, where an AE can be classified as any but the non-adversarial class or the targeted setting, where an AE has to be found within a specific target class. The scenario is then either measured by the misclassifications or targeted misclassifications, respectively. Note, that the adversarial criterion is exchangeable across different domains, such as NLP for example, which makes the BA so powerful.



```
Data: original image o, adversarial criterion $c(.)$, decision of model $d(.)$
Result: adversarial example $\tilde{\boldsymbol{o}}$ such that the distance $d(\boldsymbol{o}, \tilde{\boldsymbol{o}}) = \|\boldsymbol{o} - \tilde{\boldsymbol{o}}\|_2^2$ is minimized
initialization: $k = 0$, $\tilde{\boldsymbol{o}}^0 \sim \mathcal{U}(0, 1)$ s.t. $\tilde{\boldsymbol{o}}^0$ is adversarial;
while $k <$ maximum number of steps do
    draw random perturbation from proposal distribution $\boldsymbol{\eta}_k \sim \mathcal{P}(\tilde{\boldsymbol{o}}^{k-1})$;
    if $\tilde{\boldsymbol{o}}^{k-1} + \boldsymbol{\eta}_k$ is adversarial then
        set $\tilde{\boldsymbol{o}}^k = \tilde{\boldsymbol{o}}^{k-1} + \boldsymbol{\eta}_k$;
    else
        set $\tilde{\boldsymbol{o}}^k = \tilde{\boldsymbol{o}}^{k-1}$;
    end
    $k = k + 1$
end
```

Figure 2: The Boundary Attack Algorithm [2]

In terms of space-time complexity, the BA algorithm runs in $O(k)$ time, where k is the number of chosen iterations to generate the final AE. Note, that the duration of one unit operation is - depending on the application - probably not determined by the computation of the AE on a local machine but more by the query time on the Black-Box model, which is stored on some external server. Those query times can take up a couple of milliseconds in the worst-case, depending on the service, data volume and other factors. Note, that operations in the while loop such as setting a variable, random sampling or logical comparisons run at constant time and thus do not affect the total time complexity (see Figure 2). Since we only have to keep track of the current AE - besides the original image and a few other parameters - the BA algorithm only takes $O(1)$ space in memory.

3.2 Mathematical Foundations

3.2.1 Initialization

The setup for the BA is straightforward, when taking the mathematical formalization of Evasion Attacks into account (see Chapter 3.2). We are given the original image $\boldsymbol{o}$, an adversarial criterion $c(.)$ which depends on the domain of the attacked Black-Box model (e.g. CV, NLP, etc.) as well as the model prediction $d(.)$. The goal is to generate an AE $\tilde{\boldsymbol{o}}$, such that the distance $dist(\boldsymbol{o}, \tilde{\boldsymbol{o}}) = ||\boldsymbol{o} - \tilde{\boldsymbol{o}}||_2^2$ in l_2-norm is minimized.

Since the paper is embedded in the CV domain, it proposes the initialization of $\tilde{\boldsymbol{o}}^{\mathbf{0}}$ for the two main scenarios - the untargeted and targeted BA. For the general untargeted case, Brendel et al. propose to sample $\tilde{\boldsymbol{o}}^{\mathbf{0}}$ from a maximum entropy distribution, because it maximizes the probability of finding a random AE. Simple grey scale images already span a vector space of 255^N possible pixel configurations, where N is the number of pixels in the image and each pixel can take one of 255 possible grey values. This makes the estimation and sampling from the maximum entropy distribution difficult, which is why Brendel et al. suggest to sample $\tilde{\boldsymbol{o}}^{\mathbf{0}}$ from a simpler, constrained uniform distribution $U(0, 255)$ [2]. In the targeted setting, $\tilde{\boldsymbol{o}}^{\mathbf{0}}$ can simply be initialized with any AE which is classified as the targeted adversarial class by the Black-Box model.

3.2.2 Initial Projection

For the sake of simplicity, the following steps are explained at the example of untargeted BAs. Note, that the targeted BA only differs by the fact that we optimize within the targeted adversarial space and not in any adversarial space as for the untargeted BA. After the initialization of $\tilde{\boldsymbol{o}}^{\mathbf{0}}$, I assume the AE to be projected on the decision boundary between the non-adversarial and the adversarial pixel space. Since this step is not explicit in the paper, the BA could also simply start its random walk from $\tilde{\boldsymbol{o}}^{\mathbf{0}}$. Though, I assume a more efficient application of a search algorithm like Binary Search between $\boldsymbol{o}$ and $\tilde{\boldsymbol{o}}^{\mathbf{0}}$, to find the minimal ϵ margin by which $\tilde{\boldsymbol{o}}^{\mathbf{0}}$ can be projected on $\boldsymbol{o}$ such that $\tilde{\boldsymbol{o}}^{\mathbf{1}} = \boldsymbol{o} + \epsilon * \tilde{\boldsymbol{o}}^{\mathbf{0}}$ is still adversarial. Even though this formulation looks similar to the formalization of Evasion Attacks, note that we simply search for a minimal scaling factor ϵ and not a full perturbation image $\boldsymbol{\eta}$. The advantage of Binary Search is its space-time complexity, which runs in $O(log(n))$ time on average and $O(1)$ space when implemented as iterative algorithm and thus would not add any complexity to the overall BA algorithm.

3.2.3 Random Walk

After computing the initial projection $\tilde{\boldsymbol{o}}^{\mathbf{1}}$, the BA performs a random walk along the decision boundary between adversarial and non-adversarial image space. Each step Δ_k is directed by a proposal distribution $P(\tilde{\boldsymbol{o}}^{\boldsymbol{k}})$ to reduce the distance of the current AE $\tilde{\boldsymbol{o}}^{\boldsymbol{k}}$ to the original image $\boldsymbol{o}$. Again, we could model the proposal distribution by a maximum entropy distribution. Since estimating and sampling from this distribution is non-trivial for high-dimensional spaces, the authors suggest to use a simple iid Gaussian, i.e. the standard normal distribution $N(0, 1)$, instead [2]. A shortcoming of this approach is, that we might lose some flexibility in terms of the proposed step directions, which leaves some room for future research. When drawing directions, i.e. perturbations $\boldsymbol{\eta}$ from the standard normal distribution, we have to follow three constraints in order for the steps to be valid. First, we have to ensure that the perturbed samples still lie within the valid image domain, i.e. each grey scale pixel should lie in the range between 0 and 255.

$$\tilde{\boldsymbol{o}}_i^{k-1} + \boldsymbol{\eta}_i^k \in [0, 255] \tag{1}$$

The second constraint restricts the perturbation size of $\boldsymbol{\eta}$ to a δ-margin of the distance of the current $\tilde{\boldsymbol{o}}^{\boldsymbol{k}}$ to the original image $\boldsymbol{o}$. By this, ”overshooting” step sizes are restricted to an upper bound.

$$||\boldsymbol{\eta}^k||_2 = \delta * d(\boldsymbol{o}, \tilde{\boldsymbol{o}}^{\boldsymbol{k-1}}) \tag{2}$$

The last constraint restricts the step size towards the original image to an ϵ-margin of the distance of the current $\tilde{\boldsymbol{o}}^{\boldsymbol{k}}$ to the original image $\boldsymbol{o}$.

$$d(\boldsymbol{o}, \tilde{\boldsymbol{o}}^{\boldsymbol{k-1}}) - d(\boldsymbol{o}, \tilde{\boldsymbol{o}}^{\boldsymbol{k-1}} + \boldsymbol{\eta}^k) = \epsilon * d(\boldsymbol{o}, \tilde{\boldsymbol{o}}^{\boldsymbol{k-1}}) \tag{3}$$

From these three constraints the random walk strategy along the decision boundary can be derived. By choosing $\tilde{\boldsymbol{o}}^{\boldsymbol{k-1}}$ as the current start point for a single, exemplary iteration step Δ_k, the random walk proceeds until convergence as follows:

1. Step size δ: Draw a perturbation sample from the standard normal distribution $\boldsymbol{\eta} \sim N(0, 1)$ and add it to the current $\tilde{\boldsymbol{o}}^{\boldsymbol{k-1}}$. Clip and rescale the perturbation length by δ, such that it fulfills constraint 1 and 2.

2. Step direction: Project $\boldsymbol{\eta}^k$ on the sphere, which is drawn around $\boldsymbol{o}$ and intersecting $\tilde{\boldsymbol{o}}^{\boldsymbol{k-1}}$ (see Figure 3). This ensures, that at least the perturbation does not move away from $\boldsymbol{o}$. Step 1 & 2 are summarized as orthogonal perturbation which is represented by the first red arrow in Figure 3. Next, test whether $\boldsymbol{o}^{\boldsymbol{k}} = \tilde{\boldsymbol{o}}_{\boldsymbol{i}}^{\boldsymbol{k-1}} + \boldsymbol{\eta}^{\boldsymbol{k}}$ is still adversarial. If not, reject the sample and go back to step 1.

3. Step size ϵ: Choose the factor ϵ to scale $\boldsymbol{o}^{\boldsymbol{k}}$, such that it fulfills constraint 3.

4. Step direction: Take step along the line, connecting $\boldsymbol{o}$ and $\boldsymbol{o}^{\boldsymbol{k}}$ by some ϵ-scale. Step 3 & 4 are summarized as distance reduction towards $\boldsymbol{o}$ which is shown by the second red arrow in Figure 3. Test whether the new $\boldsymbol{o}^{\boldsymbol{k}}$ is still adversarial. If not, reject the sample and go back to step 3. Else, set the new $\tilde{\boldsymbol{o}}^{\boldsymbol{k}}$ and repeat steps 1 to 4.

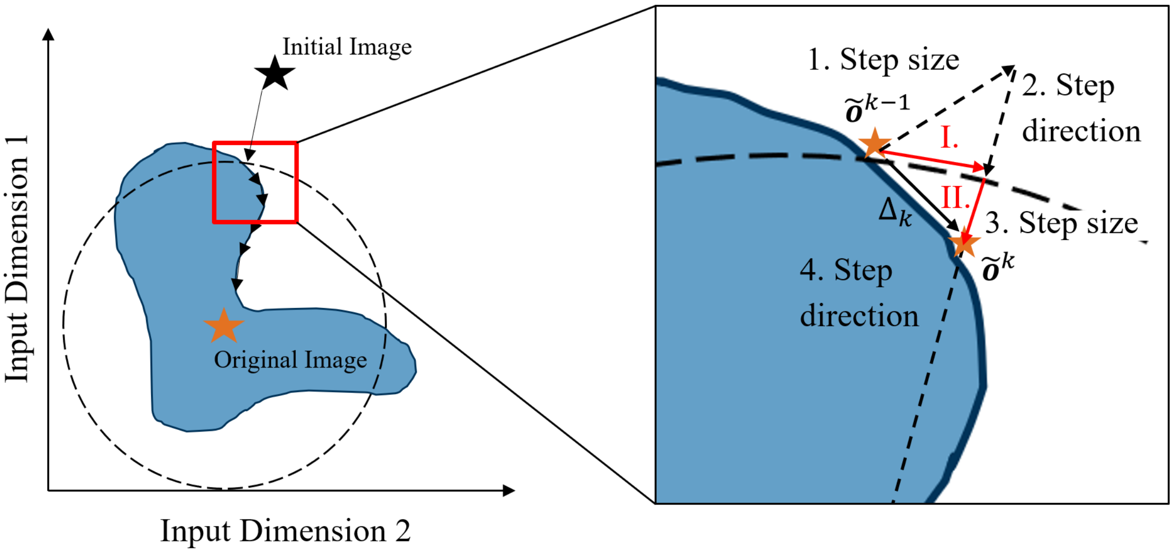


Figure 3: Intuition of the Boundary Attack [2]

3.2.4 Convergence

There exist three different termination criteria, by which the BA algorithm can be chosen to converge. Either we let it converge after k iterations or we manually perform early stopping, when the new image changes become invisible to the human visual system for example. The third way to let the algorithm terminate, is by choosing a convergence threshold for the dynamically adjusted ϵ (see Chapter 3.3). Note, that in this case no convergence guarantee is given, which is a drawback of the BA algorithm.

3.3 Dynamical Hyperparameter Adjustment

Recall, that we have the two hyperparameters δ (relative size) and ϵ (relative amount). They determine the step sizes of both, the orthogonal perturbation as well as the distance reduction towards $\boldsymbol{o}$ at each iteration (see constraint 2 and 3). We can adjust them dynamically based on the geometries of the local decision boundary.

Brendel et al. argue, that the relative size δ should be chosen such that $\sim 50\%$ of orthogonal perturbations should lie within the adversarial region, to maximize the probability of finding a valid step size. According to that, δ should be decreased, when the ratio is bigger and increased, when it is smaller. In contrast to that, the relative amount ϵ depends on the angle between the decision boundary and the virtually drawn sphere around the original image $\boldsymbol{o}$. Thus, ϵ should be increased, when the angle is big and decreased, when it is small. A naive solution for estimating the ratio to determine δ at the current $\tilde{\boldsymbol{o}}^{\boldsymbol{k}}$ is to use random point sampling in its local neighborhood. Computing the angle to determine ϵ instead, could be realised by finding the angle between the gradient of the sphere and decision boundary at the current $\tilde{\boldsymbol{o}}^{\boldsymbol{k}}$. In either case, the authors do not make a statement about their concrete computational implementation.

For this reason, I can only make assumptions about how the hyperparameters are computed efficiently. It would be unfeasible to recompute them at every iteration, due to high latencies. A more suitable solution would be to only recompute them at every k-th iteration, where k is a new hyperparameter which has to be defined. Furthermore, it is unclear how the values of δ and ϵ are obtained in the first place. The aforementioned naive solutions are imaginable, but take a lot of memory and might slow down the algorithm. I assume, that the amount of rejected samples during the random walk is stored in a separate variable, which is reset after every k-th iteration. Based on the ratio of the total rejections between two recomputations of the hyperparamters and the size of k, the new hyperparameters can be dynamically adjusted. Other solutions are imaginable.

4 Results and Evaluation

In this Chapter, the performance of the Boundary Attack is compared to a set of selected Gradient-Based Attacks. GBAs are a good benchmark in general, since they have full access to the model and its gradients. The goal is to prove the scalability of BAs to realistic threat scenarios and its competitiveness with other Evasion Attacks. Therefore, Brendel et al. suggest the median squared $L2$-distance $S_A(M)$ as evaluation metric for an attack A on a model M [2].

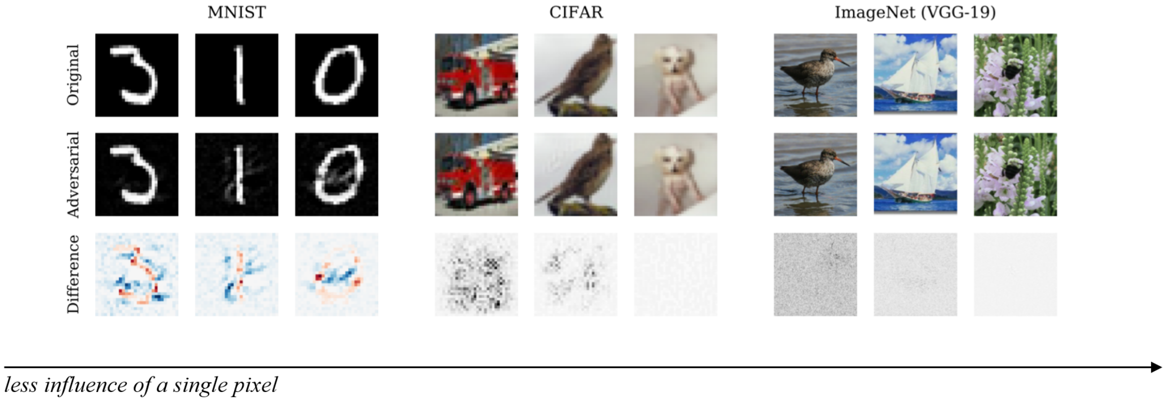
$$S_A(M) = \underset{i}{median}\ (\frac{1}{N}||\boldsymbol{\eta}_{A,M}(\boldsymbol{o_i})||_2^2)$$

The idea is to compute the average, squared l_2-norm of each image perturbation $\boldsymbol{\eta}_{A,M}(\boldsymbol{o_i})$ for a single pixel, which is found by an attack. Of all i-samples, the median length is taken, to evaluate the overall performance of each attack on a specific model.

In terms of the evaluation setup, all attacks are run on different models which are trained on three datasets. For the MNIST and CIFAR-10 dataset, Brendel et al. trained a vanilla Convolutional Neural Network, whereas on the ImageNet-1000 dataset they trained a VGG-19, ResNet-50 and Inception-v3. The attacks are evaluated in both main CV settings - untargeted and targeted attack. As a benchmark, the authors evaluate the BA against the three GBAs FGSM, DeepFool and Carlini & Wagner [2].

4.1 Untargeted Attacks

In the case of untargeted attacks, the BA clearly generates on par AEs with the other Gradient-Based Attacks. Throughout all three datasets and their corresponding models, the median squared $L2$-distance does not vary by much along all Evasion Attacks - except for slightly larger perturbations found by the FGSM. This is a remarkable outcome, since the BA only has access to the final Black-Box model decision, whereas all other methods have access to the full model gradients. Furthermore, the results show that for all attacks the median squared $L2$-distance for a single pixel perturbation decreases with increasing resolution of the images in the dataset. This means, that in general it should be easier to find an AE for high than for low resolution images, since the perturbations are less structured for the human eye. This logic is falsified by experiments, Brendel et al. ran on two models of the cloud-based Computer Vision API by Clarifai [2]. The resulting median pixel perturbations were larger by some orders of magnitude and by that not always imperceptible to the human eye. This result leaves some room for future research.



	Attack Type	MNIST	CIFAR	ImageNet VGG-19	ResNet-50	Inception-v3
FGSM	gradient-based	4.2e-02	2.5e-05	1.0e-06	1.0e-06	9.7e-07
DeepFool	gradient-based	4.3e-03	5.8e-06	1.9e-07	7.5e-08	5.2e-08
Carlini & Wagner	gradient-based	2.2e-03	7.5e-06	5.7e-07	2.2e-07	7.6e-08
Boundary (ours)	decision-based	3.6e-03	5.6e-06	2.9e-07	1.0e-07	6.5e-08

Figure 4: Comparison of untargeted Boundary Attack to Gradient-Based Attacks [2]

4.2 Targeted Attacks

As for the untargeted scenario, the BA generates on par AEs with the GBA Carlini & Wagner in the targeted setting. Note, that for both the BA and Carlini & Wagner, the median perturbation length are slightly bigger than in the untargeted scenario. While this might be due to different initializations or sampling during the attacks, it could also be due to small gaps between the non-adversarial and chosen targeted adversarial image pixel spaces. According to this, the untargeted BA searches directly along the decision boundary between non-adversarial and adversarial space. The search-space of the targeted attack instead, is restricted to the space of the adversarial target class which might not always lie in direct neighborhood of the non-adversarial class.

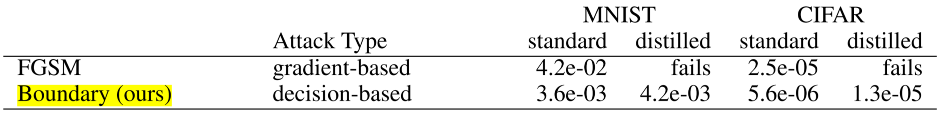
	Attack Type	MNIST	CIFAR	VGG-19
Carlini & Wagner	gradient-based	4.8e-03	3.0e-05	5.7e-06
Boundary (ours)	decision-based	6.5e-03	3.3e-05	9.9e-06

Figure 5: Comparison of targeted Boundary Attack to Gradient-Based Attacks [2]

4.3 Evaluation of the Boundary Attack

4.3.1 Advantages

The BA is suited for real-world applications and APIs, since it only requires access to the final Black-Box model prediction. Even though it needs much less information about the model than all the other attacks, it is competitive with - at least - many Gradient-Based methods. Furthermore, it introduces flexibility in terms of the chosen adversarial criterion and thus is applicable to many different domains such as CV, NLP, etc. With only two hyperparameters, which can be adjusted dynamically, it comes with far less tuning efforts than conventional Deep Learning pipelines. Last but not least, the paper proves experimentally that the BA is immune to defensive distillation which is a common gradient masking technique to defend against GBAs (see Chapter 2.2).



		MNIST		CIFAR	
	Attack Type	standard	distilled	standard	distilled
FGSM	gradient-based	4.2e-02	fails	2.5e-05	fails
Boundary (ours)	decision-based	3.6e-03	4.2e-03	5.6e-06	1.3e-05

Figure 6: Immunity of the Boundary Attack against Defensive Distillation [2]

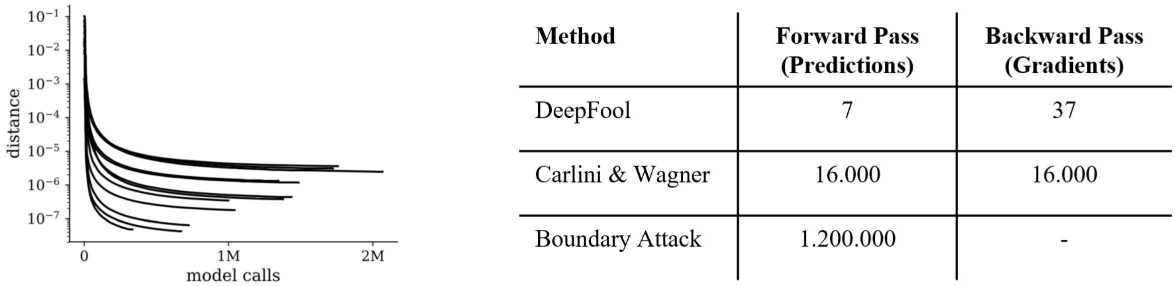
Brendel et al. argue this immunity to potentially hold for more defense techniques, without giving further experimental proofs or any mathematical intuition for why this might be the case. While at the end, this statement has to be proven, I suggest an intuition of why BAs could be immune to many types of defense strategies. Deep Neural Networks by heart are high-dimensional, discriminative function approximation algorithms, which makes them so powerful nowadays. When searching close to the decision boundary of such function approximations - as with the BA -, it should always be possible to find AEs by taking guided steps in this region. This is due to inherent nature of DNNs and thus not a bug, but more a feature as stated by Ilyas et al. [12]. The intuition is assumed to be independent of the chosen defense strategy, since DNNs always have to act as discriminators to compute a final decision, which is the single input the BA needs. Note that model randomization, as introduced in Chapter 5.2 is an experimentally proven and thus effective defense strategy, though.

4.3.2 Room for Improvement

The aforementioned advantages make the BA to appear as an almost inerrant attack strategy, if it would not require large amounts of model calls for creating minimal distance AEs or AEs in general. This is a shortcoming in terms of query inefficiency and causes real-time limitations, which can make the BA impractical for some real-world applications.

The query inefficiency can cause problems, when the attacked providers implement query-thresholds on their Black-Box models. A naive solution which I suggest for this problem is to distribute the attackers on different machines, while updating the same BA in the background. Another straightforward solution is to simply reduce the amount of necessary model calls, which is aimed by some state of the art approaches in Chapter 5.1.

As stated in Chapter 3.1, the BA runs with $O(k)$ time complexity, where the duration of a single iteration highly depends on the query time on an external Black-Box model. Assuming an average query time of around 0.2 seconds on a server and about 100,000 model calls, this sums up to a total computing time of approximately 5.5 hours, to generate a single AE. Compared to other GBAs such as DeepFool or Carlini & Wagner for example, the BA thus takes much more time to generate an AE (see Figure 7). This might become a problem, when real-time capability is an important factor for choosing the proper attack strategy. Even though the BA can take over a million model calls for creating minimal distance AEs, note that it quickly converges to sample specific plateaus after around 10,000 to 100,000 model calls already. Thereupon only small improvements are made, which are most probably invisible to the human eye and thus can be discarded in practice.



Method	Forward Pass (Predictions)	Backward Pass (Gradients)
DeepFool	7	37
Carlini & Wagner	16.000	16.000
Boundary Attack	1.200.000	-

Figure 7: Comparison for Amount of Model Calls [2]

The second, possible shortcoming of the BA is, that it might not be immune against a certain defense strategy. Both, the inadequate experimental proof as well as my suggested, vague intuition for why BAs could have a high degree of immunity, must be falsified by new defense strategies. Therefore, a state of the art defense is presented in Chapter 5.2.

5 State of the Art

5.1 Query Costs

Many of the BA follow-up papers address its query inefficiency, i.e. its large amounts of required model calls. Notable examples are the Sign-Opt Attack, which makes use of an optimization-based approach [13], qFool which is a geometry-inspired Decision-Based Attack [14] and the Biased Boundary Attack which applies a biased sampling strategy to gain efficiency from domain knowledge [15].

A mentionable example for one of the newest attacks in this field is the *HopSkipJumpAttack*, which requires much fewer queries than other state of the art DBAs [16]. The main idea of this attack is to use binary information in the decision boundary region to compute more sophisticated step directions. Its functionality is depicted in Figure 8. First, the algorithm performs a Binary Search along the connecting line between the current AE $\tilde{\boldsymbol{o}}^{\boldsymbol{k}}$ and the original image $\boldsymbol{o}$ to find $\boldsymbol{o}^{\boldsymbol{k}}$ at the decision boundary. Note, that this might be the same strategy by which the BA projects the initial AE on the decision boundary to start its random walk. Next, the algorithm estimates the gradient direction at $\boldsymbol{o}^{\boldsymbol{k}}$ based on some Boolean-valued function. This is followed by a so called "geometric progression" along the identified gradient direction to find $\tilde{\boldsymbol{o}}^{\boldsymbol{k+1}}$ in the third step. Finally, the algorithm goes back and repeats the steps multiple times until it converges.

While the HopSkipJumpAttack as well as many other of DBAs outperform the BA by 5x up to 10x its speed, still they take a couple of minutes and thousands of model

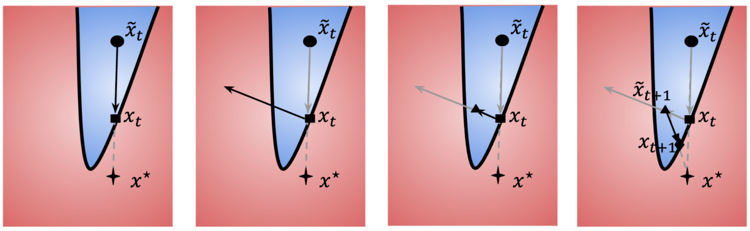


Figure 8: The HopSkipJumpAttack [16]

calls to generate a feasible AE. The bottleneck of these approaches remains in the query time, which is inherent to their threat scenario. With the attacker distribution, I suggest a possible workaround for this problem in Chapter 4.3.2. Nevertheless, future research should focus on further reducing the amount of model calls for more efficient DBAs.

5.2 Defense Strategy

An important concept, which emerged in the last years is *certified robustness* [17]. The idea is that small perturbations in the input data of a model - like an image - should only lead to small changes in the output data - like the classification of this image. According to this, a model is aimed to be invariant to data perturbations in a certified region, bounded by some l_p-norm. Much research has focused since on formal methods, such as Convex Relaxation, Randomized Smoothing, Differential Privacy etc. [17] and experimental methods [3], to realise certified robustness for different models.

Both, the perturbation size and the chosen l_p-norm under which robustness is aimed, highly depend on the specific threat scenario and are subject to the ongoing scientific debate. On top of that, researchers like Nicholas Frosst raise the need for semantic robustness, which can not always be guaranteed by certifiable robustness alone [18]. However, Carlini et al. claim, the more l_p-bounded robustness is added to a model, the better its overall robustness against AEs [19], which is also confirmed by Biggio et al. [3].

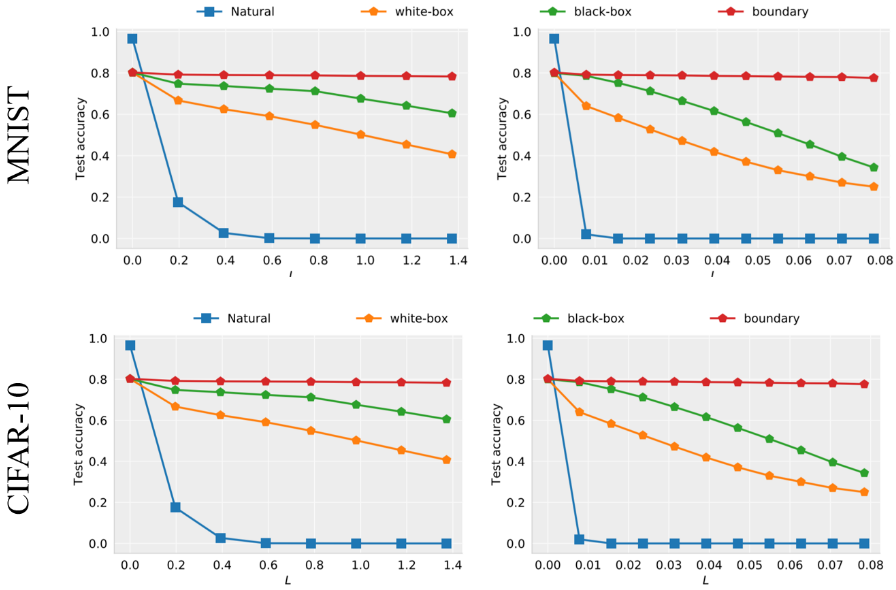


Figure 9: Results for Stability Tranining with Noise (STN) [9]

Another important claim by Carlini et al. is, that the state of the art defense strategy against gradient-free and thus DBAs are randomized models [19]. Li et al. implement this principle by using *stability training with noise* (STN) [9]. The idea of STN is leant on standard robust training as introduced in [6], by only making use of Gaussian perturbations. The results in Figure 9 show, that models trained with STN on the MNIST as well as the CIFAR-10 dataset can indeed defend against the BA for both, image perturbations measured in l_2- (left) as well as l_∞-norm (right).

It follows, that an interesting direction for future work would be to test, if one of the newest DBAs - such as the HopSkipJumpAttack - are also ineffective against the STN defense strategy or adversarial training on AEs identified by the attack itself. Therefore, I conducted small experiments with the *Adversarial Robustness Toolbox*, which can be found on Github (`https://github.com/mo12896/ASML_Seminar.git`). There I also discuss my experimental results in the context of this seminar and issues I was facing.

6 Conclusion

The focus of this work is to express the main ideas of the paper "Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models" by Brendel et al., set it in a context with the overall seminar, evaluate its advantages and disadvantages and suggest possible future work.

The authors of the paper introduce a new category in the field of Evasion Attacks - the Decision-Based Attacks. Those are compared to categories and attacks from other (seminar) papers in a comprehensive overview (see Chapter 2.2 and 4). While some of the categories like the GBAs or UAP attack require much less overall runtime, the Boundary Attack - as the newly introduced DBA - is much harder to defend against. The mathematical foundations of the BA as well as its experimental performance are given in Chapter 3 and 4. Its overall advantages and disadvantages are evaluated in Chapter 4.3.

With the BA, Brendel et al. formalize a remarkable approach, which only requires access to the final model prediction to generate competitive AEs. This addresses a more realistic threat scenario for applications, which often hide their models on external servers. The only problem of the BA from an attackers perspective, is the large amount of model calls. The paper could have put more efforts into solving this problem, since it makes the BA impractical for many scenarios without any further improvements. On the defender side, the authors claim the BA to be potentially immune against many known defense strategies. This is a strong statement, which is neither supported by adequate experimental proof or any mathematical intuition. Lastly, the lack of explanations for concrete implementations such as the initial image projection or the dynamical hyperparameter adjustment for example, make it harder for the reader to evaluate the performance of the BA. However, the BA opened up a new line of research in the field of Evasion Attacks by the time of its release and posed new questions on the robustness of ML models.

As shown in Chapter 5, the two main drawbacks of the BA - namely its runtime inefficiency as well as lacking defense strategies - are addressed by state of the art follow-up papers. Each of them points to interesting directions for future work, which I investigated within some experiments. Besides that, I suggest a naive solution for solving the query inefficiency problem of the BA by using distributed attackers to update the same BA.

Bibliography

[1] Christian Szegedy et al. “Intriguing properties of neural networks”. In: (Feb. 2014)). arXiv: 1312.6199.

[2] Wieland Brendel, Jonas Rauber, and Matthias Bethge. “Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models”. In: (Feb. 2018)). arXiv: 1712.04248.

[3] Battista Biggio and Fabio Roli. “Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning”. In: (July 2018)). DOI: 10.1016/j.patcog.2018.07.023.

[4] Kevin Eykholt et al. “Robust Physical-World Attacks on Deep Learning Models”. In: (Apr. 2018)). arXiv: 1707.08945.

[5] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. “Explaining and Harnessing Adversarial Examples”. In: (Mar. 2015)). arXiv: 1412.6572.

[6] Nicolas Papernot et al. “Practical Black-Box Attacks against Machine Learning”. In: (Mar. 2017)). arXiv: 1602.02697.

[7] Florian Tramer et al. “The Space of Transferable Adversarial Examples”. In: (May 2017)). arXiv: 1704.03453.

[8] Seyed-Mohsen Moosavi-Dezfooli et al. “Universal adversarial perturbations”. In: (Mar. 2017)). arXiv: 1610.08401.

[9] Bai Li et al. “Certified Adversarial Robustness with Additive Noise”. In: (Nov. 2019)). arXiv: 1809.03113.

[10] Ali Shafahi et al. “Universal Adversarial Training”. In: (2019)). arXiv: 1811.11304.

[11] Florian Tramer et al. “Ensemble Adversarial Training: Attacks and Defenses”. In: (Apr. 2020)). arXiv: 1705.07204.

[12] Andrew Ilyas et al. “Adversarial Examples Are Not Bugs, They Are Features”. In: (Aug. 2019)). arXiv: 1905.02175.

[13] Minhao Cheng and Thong Le. “Query-Efficient Hard-label Black-box Attack: An Optimization-based Approach”. In: (July 2018). arXiv: 1807.04457.

[14] Yujia Liu, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. “A Geometry-Inspired Decision-Based Attack”. In: (Mar. 2019). arXiv: 1903.10826.

[15] Thomas Brunner et al. “Guessing Smart: Biased Sampling for Efficient Black-Box Adversarial Attacks”. In: (May 2019)). DOI: 10.1109/ICCV.2019.00506.

[16] Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. “HopSkipJumpAttack: A Query-Efficient Decision-Based Attack”. In: (Apr. 2020)), pp. 1277–1294. DOI: 10.1109/SP40000.2020.00045.

[17] Mathias Lecuyer et al. “Certified Robustness to Adversarial Examples with Differential Privacy”. In: (May 2019)). arXiv: 1802.03471.

[18] Yao Qin et al. “Deflecting Adversarial Attacks”. In: (2020)). arXiv: 2002.07405.

[19] Nicholas Carlini et al. “On Evaluating Adversarial Robustness”. In: (Feb. 2019)), pp. 5, 13. arXiv: 1902.06705.