FaDec: A Fast Decision-based Attack for Adversarial Machine Learning

Faiq Khalid[1,*], Hassan Ali[2,*], Muhammad Abdullah Hanif[1], Semeen Rehman[1],
Rehan Ahmed[2], Muhammad Shafique[1]
[1]Technische Universität Wien (TU Wien), Vienna, Austria
Email: {faiq.khalid, muhammad.hanif, semeen.rehman, muhammad.shafique}@tuwien.ac.at
[2]National University of Sciences and Technology (NUST), Islamabad, Pakistan
Email: {rehan.ahmed, hali.msee17}@seecs.edu.pk

***Abstract*—Due to the excessive use of cloud-based machine learning (ML) services, the smart cyber-physical systems (CPS) are increasingly becoming vulnerable to black-box attacks on their ML modules. Traditionally, the black-box attacks are either *transfer attacks* requiring model stealing, or *score/decision-based gradient estimation attacks* requiring a large number of queries. In practical scenarios, especially for cloud-based ML services and timing-constrained CPS use-cases, every query incurs a huge cost, thereby rendering state-of-the-art decision-based attacks ineffective in such settings. Towards this, we propose a novel methodology for automatically generating an extremely fast and imperceptible decision-based attack called *FaDec*. It follows two main steps: (1) fast estimation of the classification boundary by combining the half-interval search-based algorithm with gradient sign estimation to reduce the number of queries; and (2) adversarial noise optimization to ensure the imperceptibility. For illustration, we evaluate FaDec on the image recognition and traffic sign detection using multiple state-of-the-art DNNs trained on CIFAR-10 and the German Traffic Sign Recognition Benchmarks (GTSRB) datasets. The experimental analysis shows that the proposed FaDec attack is 16x faster compared to the state-of-the-art decision-based attacks, and generates an attack image with better imperceptibility for a much lesser number of iterations, thereby making our attack more powerful in practical scenarios. We open-sourced the complete code and results of our methodology at https://github.com/fklodhi/FaDec.**

I. Introduction

Machine learning (ML)-based modules in smart cyber-physical systems (CPS) are vulnerable to several attacks that can generate adversarial examples for misclassification [1]–[3]. Most of these attacks *work under the white-box settings*[1] and compute the gradient of the loss function with respect to the input [4]–[12]. These gradient-based attacks can potentially be neutralized by using the gradient masking [13], defensive distillation [14], pre-processing-based defenses [15], [16], or non-differentiable classifier [17]. In practical scenarios, an adversary may only have access to the inputs and outputs of the ML model. For instance, the cloud-based ML services a offer black-box access to their trained models to the clients [18]. For such scenarios, several black-box attacks [19]–[26] have been developed, which typically perform *transfer attacks* using model stealing or *score-based gradient estimation* using complete/partial output probability vector [27]–[29]. These attacks can be nullified either by hiding the probability vector [30] (for score-based attacks), or by using a few of the defenses mentioned above for white-box attacks (for model stealing attacks).

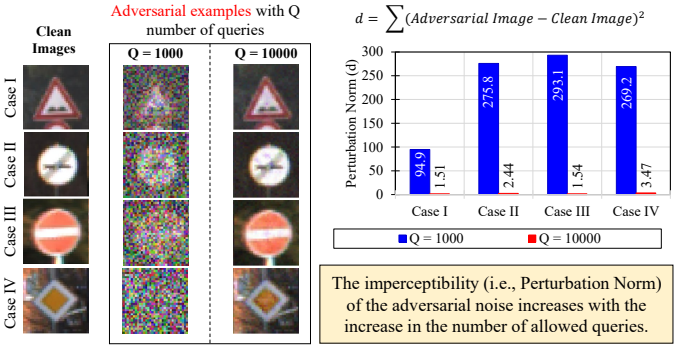

Fig. 1: Adversarial Examples generated by the decision-based attack [31] when the number of queries is restricted to 1000 and 10000. This analysis shows that the perception / visibility of the attack noise increases significantly by reducing the number of allowed queries (which is the case of real-world systems, like cloud-based ML services or resource-constrained CPS). For instance, in case III, the perturbation norm (d) of the attack noise increases, 1.54 to 293.1.

Recently, *decision-based attacks* that use the final decision of the ML model have been proposed [31]–[34]. However, most of these attacks deploy *the random search algorithm with multiple reference samples*, which *significantly increases the number of queries* to generate an imperceptible attack noise. For example, if the number of queries (Q) is restricted to 1000, the decision-based attack [31] generates the adversarial examples with a highly perceptible noise, as shown by our experimental analysis in Fig. 1. If the number of queries (Q) is increased to 10000, it generates the adversarial examples with imperceptible noise, see Fig. 1. Similarly, other so-called query-efficient score/decision-based attacks require more than 8000 queries [18]. In a practical scenario, especially in cloud-based ML services and resource and timing-constrained CPS systems (like autonomous vehicles), every query incurs with a huge cost, thereby requiring a much faster attack compared to the state-of-the-art decision-based attacks. These observations lead to the following key research question, as targeted in this paper: *how to design a resource-efficient (in terms of a reduced number of queries) attack methodology to automatically generate an attack image very fast while ensuring the imperceptibility*[2]*?*

*Faiq Khalid and Hassan Ali have equal scientific contributions.

[1]Note, all the white-box adversarial attacks can be implemented in black-box settings by combining it with model stealing attacks.

[2]Imperceptibility is ensured by maximizing the Structural Similarity Index (SSIM) and Cross Co-relation Coefficient (CC), and by minimizing the Perturbation Norm (d).

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

A. Novel Contributions and Concept Overview

To address the above research question, we propose a novel methodology to perform a Fast Decision-based (FaDec[3]) attack (see Fig. 2), which employs the following two key techniques:

1) To significantly reduce the number of queries, our methodology employs *an iterative half-interval search for finding a sample image close to the classification boundary* (**Section III**). The reason for choosing the iterative half-interval search algorithm is that it requires only one reference sample, and it converges much faster compared to the random iterative search.
2) To maximize the imperceptibility, an optimization algorithm (**Section IV**) is proposed that *combines the half-interval search algorithm with a distance-based gradient sign estimation* to identify the adversarial example close to the classification boundary.

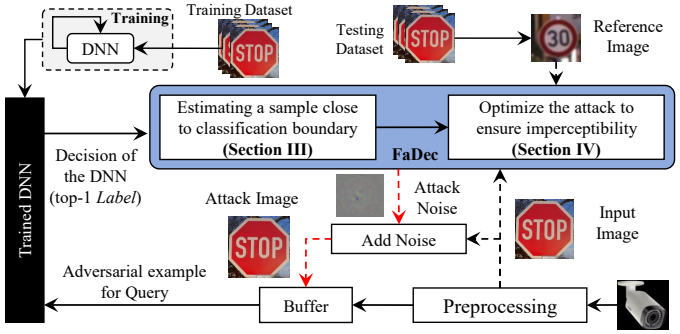

Fig. 2: Overview of our methodology to automatically generate the FaDec Attack. Novel contributions are shown in "Blue" box.

To illustrate the effectiveness of FaDec, we evaluate it for the image recognition and traffic sign detection using multiple state-of-the-art DNNs (see details in the experimental setup, Section V-A), available in an open-source library (CleverHans [36]), trained on CIFAR-10 and the German Traffic Sign Recognition Benchmarks (GTSRB) datasets. Our experimental results show that the proposed methodology is 16x faster compared to the state-of-the-art decision-based attacks [31][37], in successfully generating imperceptible adversarial examples. Our results show that, on average, the perturbation norm of the adversarial images w.r.t. their corresponding source images is decreased by 96.1%, while their SSIM and CC w.r.t. the corresponding clean images are increased by 71.7% and 20.3%, respectively.

Open-Source Contributions: We have released our complete code and configurations, for reproducible research, at https://github.com/fklodhi/FaDec.

II. Proposed Methodology for Generating the FaDec Attack

The goal of our methodology is to generate the minimum noise perturbation that is required to map an input image to a targeted "incorrect" class (for a targeted misclassification attack), or to ensure a "random" misclassification (for an un-targetted attack) with the minimum possible number of queries. Fig. 3 shows the complete step-by-step flow of our

[3]It is the same with renaming RED-Attack [35]

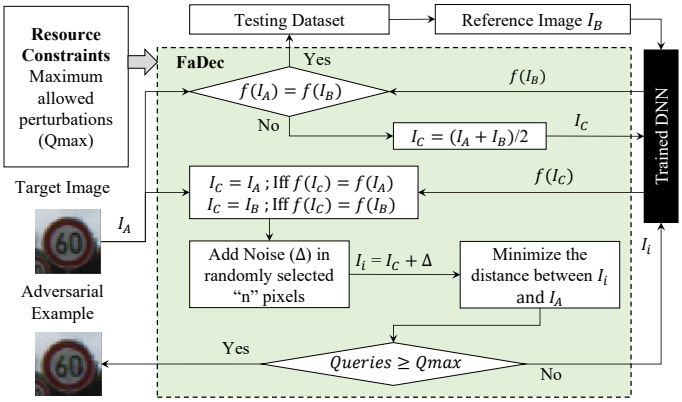

Fig. 3: Operational flow with detailed steps of our methodology to perform a fast decision-based attack (FaDec).

methodology to generate the FaDec attack, as explained below, while Fig. 4 explains the flow with the help of a pictorial example.

1) It selects a *reference image (I_B)* from the input (i.e., camera) whose output label is different from the *target image (I_A)* (in case of un-targeted misclassification), or equal to a specific label (in case of targeted misclassification). Using I_B and I_A, it performs the iterative half-interval search to find a sample I_i on the classification boundary (see Step 1 in Fig. 4). Here, *"on the classification boundary"* means that the distance of the sample from the classification boundary is within a tolerable range δ_{min}.
2) Afterward, it introduces perturbations in the sample I_i such that the output label of updated perturbed image I_{ibe} is different from the I_A (in case of un-targeted misclassification) or equal to a specific label (in case of targeted misclassification), see Step 2 in Fig. 4.
3) Then, it computes the gradient sign by comparing the distance of the perturbed sample I_{ibe}; from the target sample I_A with the distance of the previously perturbed sample I_i from the target sample I_A (see Step 3 in Fig. 4). *We choose to estimate the gradient sign instead of gradient because it requires only a single query and guides the half-interval search algorithm to search in the correct direction.* Moreover, estimating the complete gradient just to guide the search algorithm increases the complexity and thereby the number of queries.
4) The proposed methodology then again performs the half-interval search using the target image I_A and the perturbed sample I_i (see Step 4 in Fig. 4).
5) This process is repeated until distance of the perturbed sample I_i from the target image I_A is within the tolerable range (defined by Δ_{max}), or when the number of queries is equal to the maximum allowed number of queries ($Qmax$), see Step 5 in Fig. 4. The attack flow of our methodology is formally given by Algo. 1.

A. Mathematical Formulation of FaDec

To determine the adversarial example, we use one of the most commonly used cost function defined by the CW attack [8].

$$cost = c \times (f(X_{adv}) - f(X_{target}))^2 + \sum (X - X_{adv})^2 \quad (1)$$

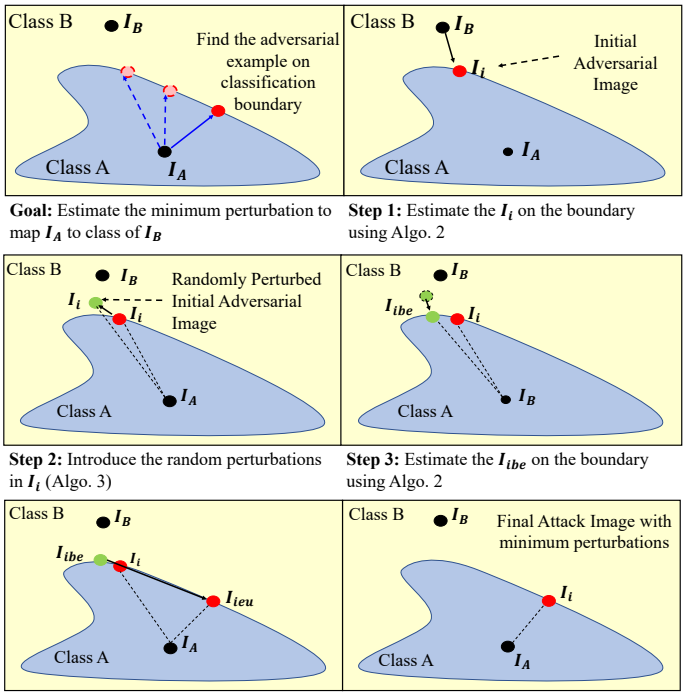

Fig. 4: An abstract example to show the step-by-step procedure of our proposed methodology to perform the FaDec attack. Note, in the figure, I_A, I_B, and I_i represent the target image from class A, reference image from targeted label (i.e., class B) and adversarial image, respectively. In case of an un-targeted attack, I_B, represents the reference image from any other class except the class A.

Where, X_{adv}, X_{target} and c represent the adversarial image, targeted image and constant, respectively. The reason behind choosing this cost function is that it minimizes the difference from target image ($(f(X_{adv}) - f(target))^2$) and magnitude of perturbation ($\sum(\Delta_x)^2$), simultaneously. However, in our attack the function (f) is discrete and differentiable?. Therefore, we reformulate the above cost function as,

$$cost = c \times (f(X_{adv}) != f(X_{target})) + \sum (X - X_{adv})^2 \quad (2)$$

In the cost function, larger value of c increases the converging time because it leads to larger value of cost function that eventually increases the time required to compute the adversarial example. Therefore, in this cost function the value of c should be close to 1. For ensuring the convergence of the cost function, we propose use the following functions to approximate the gradient of the cost function.

If the current adversarial example does not belong to target class then the gradient of the cost function is computed as:

$$\frac{\partial cost}{\partial X_{adv}} = X_{adv} - X_{target} \quad (3)$$

However, if the current I_i belongs to the target class, the gradient of the cost function cost computed as:

$$\frac{\partial cost}{\partial X_{adv}} = 2 \times (X_{adv} - X) \quad (4)$$

After computing the gradients of the cost function, the new adversarial instance is computed as:

$$X_{adv,new} = X_{adv,old} - \alpha \times \frac{\partial cost}{\partial X_{adv}} \quad (5)$$

In state-of-the-art and above mentioned gradient estimation function, the gradient is estimated in a linear manner, either towards the target example or towards the source example until we reach the boundary. This linear estimation increases the time required to compute the required adversarial example. To address this issue, we propose to use the half-interval search, as shown in Algorithm 2 and Algorithm 3. Moreover, linear estimation, either towards the target example or towards the source example, will cause infinite oscillations at the transition of the boundary. To counter this problem, we redefine our cost function for the region characterized by the δ_{min} distance of each pixel to the boundary. The new cost function is defined as

$$cost = \sum (X_{adv} - X)^2 \quad (6)$$

We optimize this new cost function using stochastic Zeroth-Order Optimization. First, we randomly select n number of pixels in the X_{adv} and introduce random perturbations in the selected pixels to compute $\bar{X}_{adv}$. The zeroth-order gradient is,

$$\frac{\partial cost}{\partial X_{adv}} = \frac{\sum (X_{adv} - X)^2 - \sum (\bar{X}_{adv} - X)^2}{X_{adv} - \bar{X}_{adv}} \quad (7)$$

$$X_{adv,new} = X_{adv,old} - \lambda \times \frac{\partial cost}{\partial X_{adv}} \quad (8)$$

The magnitude of "λ" is adjusted efficiently to make a jump that brings the adversarial examples closest to the source example.

In the following, we explain the proposed techniques for "estimating the sample on the classification boundary" (**Section III**) and "optimizing the attack noise" (**Section IV**).

Algorithm 1 Methodology to perform FaDec Attack

Input:
I_A = Target image;
I_B = Reference image;
Δ_{max} = Maximum Square L2-Distance tolerable;
Q_{max} = Maximum allowed queries;
n = Number of pixels to perturb;
θ = Relative Perturbation in each pixel;
δ_{min} = Max. Allowed Perturbation;
Output:
I_i = Adversarial Image;
1: Compute $I_i = I_{ibe}$ using Algo. 2;
2: **repeat**
3: Update $I_i = I_{ige}$ and compute g using Algo. 3;
4: Update $I_i = I_{ieu}$ using Algo. 4;
5: Compute $I_i = I_{ibe}$ using Algo. 2;
6: **until** $(\sum(I_i - I_A)^2 > \Delta_{max})$ & $Q \leq Q_{max}$

III. Estimating the Sample I_i on the Classification Boundary

We first formulate the problem of estimating the sample on classification boundary in the following goal.

Goal: Let I_A, I_B and δ_{min} be the source image (class: A), reference image (class: other than A) and maximum allowed estimation error. *The goal of this algorithm is to find a sample*

Algorithm 2 Estimating a Sample on Classification Boundary

Input:
I_A = Target image; I_B = Reference image;
δ_{min} = Max. Allowed Perturbation;
Output:
I_{ibe} = Adversarial Image;
1: Select a sample Adversarial Image (I_i)
2: $I_{ibe} = \frac{I_A + I_B}{2}$;
3: **repeat**
4: $k = f(I_{ibe})$; $Q = Q + 1$;
5: **if** $f(I_A) \neq k$ **then**
6: $I_B = I_{ibe}$;
7: **else**
8: $I_A = I_{ibe}$;
9: **end if**
10: $\delta = max(I_A - I_{ibe})$;
11: **until** $\delta \leq \delta_{min}$

I_i which has tolerable distance (less than δ_{min}) from the classification boundary and has a label different from the source image. Mathematically, it can be defined as:

$$\exists\ I_i : f(I_i)\ \neq\ f(I_A)\ \wedge\ max(I_i - I_A)\ \leq\ \delta_{min} \tag{9}$$

To generate the appropriate I_i, the proposed algorithm first finds the half way point I_i between the source image (I_A) and the reference image (I_B) by computing the average of the two, and then replaces I_A or I_B with I_i depending upon the class in which I_i falls (see line 2 in Algo. 2). For example, if the label of the half way point I_i is class A then algorithm replaces I_A with I_i, and if its label is not A then the algorithm replaces I_B with I_i (see lines 4-9 Algo. 2). The algorithm repeats this process until the maximum distance of I_i from the I_A is less than δ_{min}, and while ensuring the $f(I_i)\ \neq f(I_A)$. The proposed boundary estimation can be used for targeted attack if we choose the reference image from the target class.

IV. Optimize the Attack Noise

To ensure the imperceptibility of the attack noise on the sample I_i (output of the boundary estimation), we propose to incorporate the *"adaptive update in the zeroth order stochastic algorithm"*. We first formulate the problem of optimizing the attack noise in the following goal.

Goal 2: Let I_A, I_B, δ_{min} and I_i be the source image (class: A), reference image (class: other than A), maximum allowed estimation error and perturbed image, respectively. *The goal of this algorithm is to minimize distance of I_i from I_{A2} while ensuring that it has label different than the source image.* Mathematically, it can be defined as:

$$\forall\ I_i\ min(I_i - I_A) : f(I_i)\ \neq f(I_A) \tag{10}$$

To achieve this goal, we first identify the gradient sign g to guide the half-interval search in appropriate direction for identifying the sample I_i on the local minima (with respect to the distance from the target image I_A) of the classification boundary. For example, if the gradient sign is negative (as illustrated by the green arrow in Fig.5) then we continue moving in the same direction; otherwise we change the direction (as illustrated by the red arrow in Fig.5), as shown in lines 5-6 of Algo. 3.

Once the direction is identified, the next key challenge is to select the appropriate jump size λ. Therefore, to select the λ,

Algorithm 3 Gradient Sign Estimation

Input:
I_A = Target image;
I_{ibe} = Output of Algorithm 2;
n = Number of pixels to perturb;
θ = Perturbation in each pixel;
Output:
I_{ige} = Adversarial Image; g = Gradient Sign;
1: Define I_0 of size I_i and set all its values to 0;
2: Randomly select n pixels in I_0 and set their values to the maximum value of a pixel;
3: $I_{ige} = I_{ibe} + \theta \times I_0$; $g = -1$;
4: Update $I_ige = I_ibe$ using Algo. 2;
5: **if** $\left(\sum(I_{ige} - I_A)^2\right) > \left(\sum(I_{ibe} - I_A)^2\right)$ **then**
6: Compute $g = 1$;
7: **end if**

Algorithm 4 Adaptive Update to Find I_i on the Classification Boundary with Minimum Distance from I_A

Input:
I_A = Target image;
I_{ibe} = Output of Algorithm 2;
I_{ige} = Output of Algorithm 3;
g = Gradient Sign;
j = Maximum Jump;
Output:
I_{ieu} = Attack Image;
1: $\Delta = -g \times (I_{ige} - I_{ibe})$;
$\lambda = j$;
2: **repeat**
3: $I_{ieu} = I_{ibe} + \lambda \times \Delta$;
4: Update $I_ige = I_ibe$ using Algo. 2;
5: $\lambda = \frac{\lambda}{2}$;
6: **until** $(\sum(I_{ieu} - I_A)^2 > \sum(I_{ibe} - I_A)^2)$ & $(\lambda > 0.004)$

we propose an algorithm (Algo. 4) to efficiently find the local minima on classification boundary. The proposed algorithm first initialize the jump size λ with the maximum jump size j and then reduces it by half in each iteration until the distance of the perturbed sample I_{ieu} from I_A is less than or equal to the the distance of the sample I_{ibe} form I_A.

Then to move I_{ieu} on the classification boundary, it applies Algo. 2 on the sample I_{ieu} and finds the updated perturbed sample I_{ibe}. Finally, algorithms 3, 4 and 2 are repeated until the FaDec attack finds the sample I_i on the classification boundary with minimum distance from I_A, as illustrated in Algo. 1 and also illustrated in Steps 3 to 5 in Fig. 4.

V. Results and Discussions

A. Experimental setup

To demonstrate the effectiveness of the proposed RED-Attack, we evaluate several un-targeted attacks using the

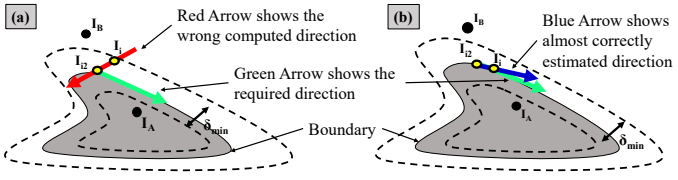

Fig. 5: A Pictorial example to illustrate the process of identifying the gradient sign and I_i on the classification boundary with minimum distance from I_A.

following experimental setup.

1) **Datasets:** CIFAR-10 (70% Test Accuracy), GTSRB (95.2% Test Accuracy)
2) **DNN for CIFAR-10:** Conv2D(64, 3x3) - Conv2D(64, 3x3) - Conv2D(128, 3x3) - Conv2D(128, 3x3) - Dense(256) - Dense(256) - Dense(10) - Softmax()
3) **DNN for GTSRB:** Lambda(lambda p: p/255.0 - 0.5) - Conv2D(3, 1x1) - Conv2D(16, 5x5, (2, 2)) - Conv2D(32, 3x3) - MaxPool2D((2, 2), (2, 2)) - Conv2D(64, 3x3) - Conv2D(128, 3x3) - Flatten() - Dropout(0.5) - Dense(128) - Dropout(0.5) - Dense(43) – softmax()
4) **Training parameters for DNNs:** Epoch = 15; Batch Size = 128; Optimizer = Adam; Learning Rate = 0.0001; Decay = 1×10^{-6}.

B. Evaluation Parameters

For comprehensive evaluation, we use the following parameters. Their value ranges considered in our experiments are shown in Table I.

1) δ_{min} *measures the maximum tolerable error distance* between the actual classification boundary and the estimated boundary, as illustrated in Fig. 5. We computed it as the maximum distance between the two samples in half-interval search (line 10 of Algorithm 2).
2) n *defines the number of pixels, randomly selected to be perturbed in each iteration* in order to estimate the gradient of the distance of the adversarial example from the source example (Algorithm 3).
3) θ defines the magnitude of the noise added in each of n randomly selected pixels relative to the maximum value, a pixel can have.

C. Evaluation Metrics for Imperceptibility

To evaluated the imperceptibility of the adversarial image, we use the following metrics.

1) **Perturbation Norm (d)** is defined as the mean square distance between the adversarial image and the clean image. Note: for high imperceptibility, the value of the d should be close to "0".
2) **Cross Co-relation Coefficient (CC)** is defined as the degree of probability that a linear relationship exists between two images. To compute the CC, we used the Pearson's correlation coefficient [38] from python library "skimage". Note: for high imperceptibility, the value of the CC should be close to "1".
3) **Structural Similarity Index (SSIM)** is defined as the perceptual similarity between two images. To compute the SSIM, we used a built-in function of from python library "skimage". This function computes the SSIM based on contrast, luminance and structure comparison [39]. Note: for high imperceptibility, the value of the $SSIM$ should be close to "1".

TABLE I: Values of Evaluation Parameters for Experimental Analysis

Evaluation Parameter	Values used in Experiments	
	Range	Values
δ_{min}	1 to 15	1, 5, 10, 15
n	5 to 50	5, 10, 30, 50
θ	0.0196 to 0.1962	0.0196, 0.0392, 0.1176, 0.1962

Fig. 6: Performing the Targeted and Un-targeted Attacks on the classifier using only the decision provided by the classifier. (Top 10 rows) Adversarial examples found for the targeted attacks. Correct labels are shown top in black on top. The target labels are shown in red on left. (Last Row) Adversarial examples found for un-targeted attacks. Maximum number of queries = 10^5

Adversarial examples found for the targeted and un-targeted attack scenarios against the black-box classifier are shown in Figure 6. Three images have been highlighted. We find that these images are similar to the images used as initial targets for them. In other words, our black-box attack fails to find a satisfactory adversarial example in these cases. However, we repeat the experiment for the same source and target image several times and find that the algorithm never fails again. There may be many reasons as to why it failed for the first time. The simplest one being a rare encounter of the local minima or the saddle point.

D. Evaluation and Discussion

In this section, we evaluate our methodology w.r.t. *d*, *CC* and *SSIM*, to illustrate different design trade-offs, and impact of different parameters.

Number of Queries: Figs. 7, 8 and 9 show that the proposed FaDec attack converges very fast to achieve the desired imperceptibility. By analyzing these results, we observed that FaDec requires less than 500 queries to generate the imperceptible adversarial noise, which is almost 16x less than the number of queries required by the state-of-the-art decision-based attack [31]. For comprehensive evaluations, we also analyzed the effects of different parameters, i.e., δ_{min}, n and θ, on the convergence of FaDec attack, as shown in Fig. 7 and 8. By analyzing these results, we make the following key observations:

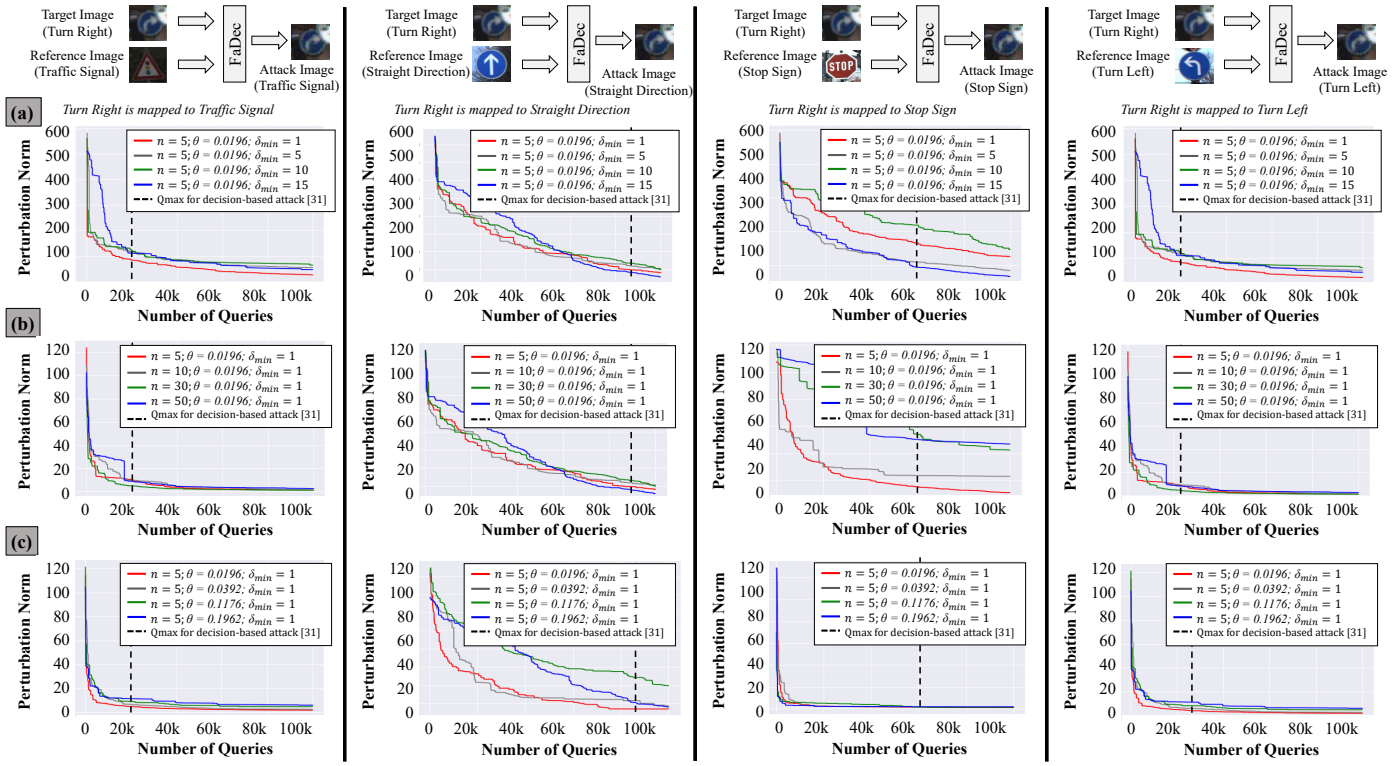

*Fig. 7: Effects of δ_{min}, n and theta on the convergence of FaDec attack (perturbation norms vs. the number of queries) for state-of-the-art CNNs trained on the **GTSRB dataset**. The dotted vertical lines in the figure show the number of queries required to converge the state-of-the-art decision-based attack [31]. These analyses show that in all the cases, our FaDec attack using appropriate values of δ_{min}, n and theta converges 16x faster than the state-of-the-art decision-based attack [31].*

1) As δ_{min} increases, the quality of the adversarial example at a given query count decreases, due to the increase in its distance from the source example, as shown in Figs. 7(a) and 8(a). The reason of this behavior is that the larger value of δ_{min} results in an imprecise boundary point, which in turn may result in an incorrect value of the estimated gradient direction, as illustrated in Fig. 5. However, the smaller value of δ_{min} results in a correct gradient direction.
2) A larger value of n, initially results in a faster convergence, as shown in Figs. 7(b) and 8(b). The reason is that, we only need to estimate the overall trend of the boundary at the initial stages. Estimating the updated direction for the adversarial example by perturbing a large number of values at once helps to achieve better results. However, a larger value of n is highly vulnerable to divergence as the attack progresses, as shown in Figs. 7(b) and 8(b). This observation suggests that the attack can significantly be improved by changing the number of pixels perturbed, as the algorithm progresses in an adaptive manner.
3) Similar trend is observed with the changes in θ because large perturbations lead to higher value of δ_{min}. This in turn helps the algorithm to initially converge faster. However, small values of θ give a more stable convergence towards the solution, as shown in Figs. 7(c) and 8(c).

E. Key Insights

1) Generally, the effect of changing evaluation parameters on the perturbation norm of the adversarial example is almost similar for the GTSRB and the CIFAR-10 datasets.
2) We observe that the adversarial examples for un-targeted attack against the GTSRB dataset converge much faster as compared to the CIFAR-10 dataset, as shown in Figs. 7 and 8. We attribute this to the much larger number of classes in the GTSRB dataset as compared to that in the CIFAR-10 dataset.
3) As was observed in the case of CIFAR-10, the attack can significantly be improved by adaptively changing the evaluation parameters as the attack progresses see Figs. 7(c) and 8(c).

Imperceptibility: Fig. 9 shows the evolution of the adversarial example with respect to number of queries. The adversarial images generated in the first few iterations are not imperceptible, even not recognizable (see adversarial images after 20 queries in Fig. 9), but over time the optimization algorithm achieves the imperceptibility. It can also be observed that the adversarial noise is not visible after 200 queries, which shows the efficiency of our methodology.

VI. Comparison with the State-of-the-art Attack

We compare our results with the state-of-the-art decision-based attack [31] based on its implementation provided in an open-source benchmark library, FoolBox [37]. We limit the maximum number of queries to 1000 and evaluate our attack for different values of δ_{min}, n and θ. To compare our results with the decision-based attack [31], we use three evaluation metrics, i.e., CC, Perturbation Norm (the squared

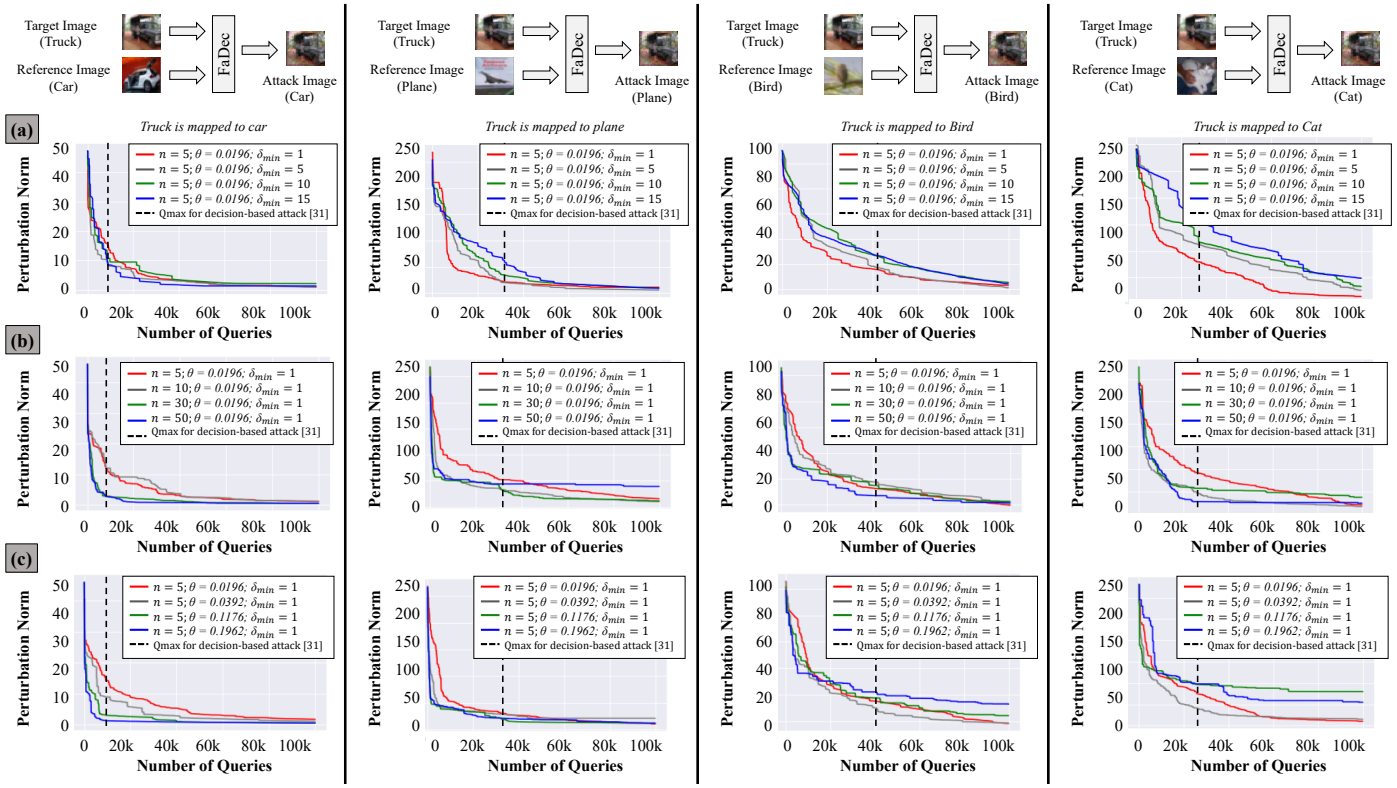

*Fig. 8: Effects of δ_{min}, n and theta on the convergence of FaDec attack (perturbation norms vs. the number of queries) for state-of-the-art CNNs trained on the **CIFAR-10 dataset**. The dotted vertical lines in the figure show the number of queries required to converge the state-of-the-art decision-based attack [31]. These analyses show that in all the cases, our FaDec attack using appropriate values of δ_{min}, n and theta converges 16x faster than the state-of-the-art decision-based attack [31].*

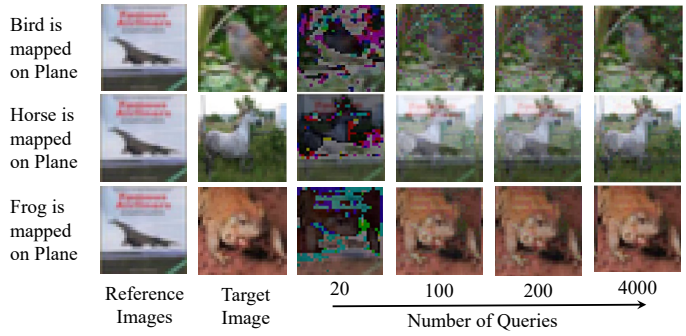

Fig. 9: The adversarial examples generated by FaDec using CIFAR-10 dataset w.r.t. the query count.

L2-Norm) and SSIM of the adversarial image w.r.t the source image.

Fig. 10 shows that the adversarial examples produced by FaDec attack are significantly superior to those produced by the decision-based attack. The reason is the binary stepping while searching for the boundary point, and the efficient adaptive update process while computing a new adversarial example. For example, in the Case C, the perturbation norm after 1000 queries is almost 3 times higher than the different settings of FaDec attack. Similarly, the achieved CC and SSIM by FaDec attack is almost 2.5 times higher than that by the decision-based attack of [31].

Note, in the long run, if the query efficiency is not much of a concern or the number of maximum queries is limited to 10^5 instead of 10^3, the adversarial examples found by the decision-based attacks can be better than those found by the FaDec attack. However, we would like to emphasize that the goal of FaDec attack is to propose an efficient attack with very few queries, such that it can be employed in practical scenarios, e.g., cloud-based ML services and resource-constrained CPS, as discussed in Section I.

VII. Conclusion

We proposed a novel methodology to perform a Fast Decision-based (FaDec) attack novel. It utilizes a half-interval search-based algorithm to estimate the classification boundary, and an efficient adaptive update mechanism to boost the convergence of an adversarial example for decision-based attacks, in query limited settings. We evaluated it for the CIFAR-10 and the GTSRB datasets using multiple state-of-the-art DNNs. FaDec is 16x faster compared to state-of-the-art decision-based attack [31]. Furthermore, we showed that for 1000 queries, the state-of-the-art decision-based attack is unable to find an imperceptible adversarial example, while the FaDec attack finds a sufficiently imperceptible adversarial example. On average, the perturbation norm of adversarial images (from their corresponding source images) is decreased by 96.1%, while the values of their SSIM and CC (with respect to the corresponding clean images) are increased by 71.7% and 20.3%, respectively. The complete code and results of our methodology are available at https://github.com/fklodhi/FaDec.

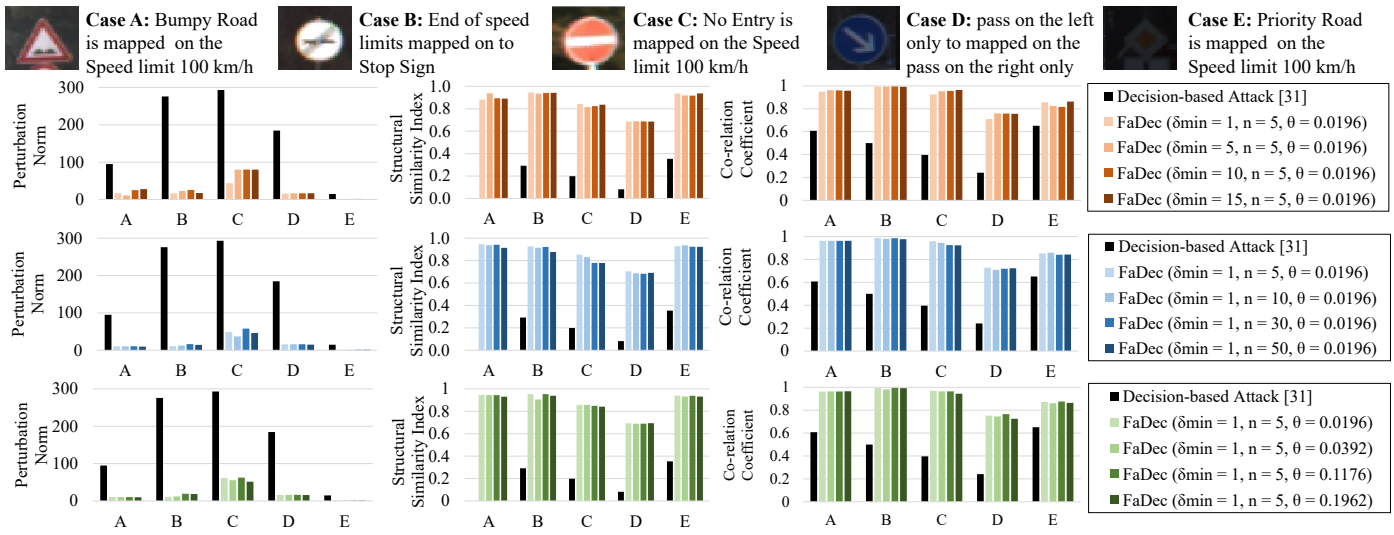

Fig. 10: Comparison of FaDec for different cases with the state-of-the-art decision-based attack [31] for different values of δ_{min}, n and θ. The maximum number of allowed queries Q_{max} is fixed to 1000. The analyses show that FaDec generates better imperceptibility of adversarial noise as compared to state-of-the-art decision-based attack [31]. For example, in the case the decision-based attack [31] for Case C, the values of perturbation norm, CC and SSIM are approximately 2.5 times higher than the different settings of FaDec.

ACKNOWLEDGEMENT

This work was partially supported by the Erasmus+ International Credit Mobility (KA107).

REFERENCES

[1] M. A. Hanif et al., "Robust machine learning systems: Reliability and security for deep neural networks," in ***IEEE IOLTS***, 2018, pp. 257–260.
[2] F. Khalid et al., "Security for machine learning-based systems: Attacks and challenges during training and inference," in ***IEEE FIT***, 2018, pp. 327–332.
[3] M. Shafique et al., "An overview of next-generation architectures for machine learning: Roadmap, opportunities and challenges in the iot era," in ***IEEE DATE***, 2018, pp. 827–832.
[4] A. Madry et al., "Towards deep learning models resistant to adversarial attacks," *arXiv:1706.06083*, 2017.
[5] A. Kurakin et al., "Adversarial examples in the physical world," *arXiv:1607.02533*, 2016.
[6] S.-M. Moosavi-Dezfooli et al., "Deepfool: a simple and accurate method to fool deep neural networks," in *IEEE* ***CVPR***, 2016, pp. 2574–2582.
[7] N. Papernot et al., "The limitations of deep learning in adversarial settings," in *IEEE* ***EuroS&P***, 2016, pp. 372–387.
[8] N. Carlini et al., "Towards evaluating the robustness of neural networks," in *IEEE* ***S&P***, 2017, pp. 39–57.
[9] F. Khalid et al., "TrISec: Training Data-Unaware Imperceptible Security Attacks on Deep Neural Networks," in *IEEE* ***IOLTS***, 2019, pp. 188–193.
[10] M. Shafique et al., "Robust machine learning systems: Challenges, current trends, perspectives, and the road ahead," *IEEE* ***D&T***, vol. 37, no. 2, pp. 30–57, 2020.
[11] J. J. Zhang et al., "Building robust machine learning systems: Current progress, research challenges, and opportunities," in *ACM/IEEE DAC*, 2019, pp. 1–4.
[12] A. Marchisio et al., "Deep learning for edge computing: Current trends, cross-layer optimizations, and open research challenges," in ***IEEE ISVLSI***. IEEE, 2019, pp. 553–559.
[13] I. Goodfellow, "Gradient masking causes clever to overestimate adversarial perturbation size," *arXiv:1804.07870*, 2018.
[14] N. Papernot et al., "Distillation as a defense to adversarial perturbations against deep neural networks," in *IEEE* ***S&P***, 2016, pp. 582–597.
[15] F. Khalid et al., "QuSecNets: Quantization-based Defense Mechanism for Securing Deep Neural Network against Adversarial Attacks," in *IEEE* ***IOLTS***, 2019, pp. 182–187.
[16] H. Ali et al., "SSCNets: Robustifying DNNs using Secure Selective Convolutional Filters," *IEEE* ***D&T***, vol. 37, no. 2, pp. 58–65, 2020.
[17] J. Lu, T. Issaranon, and D. Forsyth, "Safetynet: Detecting and rejecting adversarial examples robustly," in *IEEE* ***ICCV***, 2017, pp. 446–454.
[18] A. Nitin Bhagoji et al., "Practical black-box attacks on deep neural networks using efficient query mechanisms," in *IEEE* ***ECCV***, 2018, pp. 154–169.
[19] N. Papernot et al., "Practical black-box attacks against machine learning," in *ACM* ***Asia CCS***, 2017, pp. 506–519.
[20] J. Gao et al., "Black-box generation of adversarial text sequences to evade deep learning classifiers," in *IEEE* ***SPW***, 2018, pp. 50–56.
[21] P.-Y. Chen et al., "Recent progress in zeroth order optimization and its applications to adversarial robustness in data mining and machine learning," in ***ACM SIGKDD***, 2019, pp. 3233–3234.
[22] S. N. Shukla et al., "Black-box adversarial attacks with bayesian optimization," *arXiv:1909.13857*, 2019.
[23] C. Guo et al., "Simple black-box adversarial attacks," *arXiv:1905.07121*, 2019.
[24] P. Zhao et al., "On the design of black-box adversarial examples by leveraging gradient-free optimization and operator splitting method," in *IEEE ICCV*, 2019, pp. 121–130.
[25] C. et al., "Improving black-box adversarial attacks with a transfer-based prior," in *NuerIPS*, 2019, pp. 10 932–10 942.
[26] F. Suya, J. Chi, D. Evans, and Y. Tian, "Hybrid batch attacks: Finding black-box adversarial examples with limited queries," *arXiv preprint arXiv:1908.07000*, 2019.
[27] N. a. Narodytska et al., "Simple black-box adversarial perturbations for deep networks," *arXiv:1612.06299*, 2016.
[28] C. et al., "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *ACM* ***AISEC***, 2017, pp. 15–26.
[29] L. Huang et al., "Adversarial machine learning," in *ACM workshop on Security and artificial intelligence*, 2011, pp. 43–58.
[30] F. Tramèr et al., "Ensemble adversarial training: Attacks and defenses," *arXiv:1705.07204*, 2017.
[31] W. Brendel et al., "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," *arXiv:1712.04248*, 2017.
[32] Y. Dong et al., "Efficient decision-based black-box adversarial attacks on face recognition," *arXiv:1904.04433*, 2019.
[33] J. Chen et al., "Boundary attack++: Query-efficient decision-based adversarial attack," *arXiv:1904.02144*, 2019.
[34] M. Cheng et al., "Query-efficient hard-label black-box attack: An optimization-based approach," *arXiv:1807.04457*, 2018.
[35] F. Khalid et al., "RED-Attack: Resource efficient decision based attack for machine learning," *arXiv preprint arXiv:1901.10258*, 2019.
[36] N. Papernot et al., "cleverhans v1. 0.0: an adversarial machine learning library," *arXiv:1610.00768*, vol. 10, 2016.
[37] R. et al., "Foolbox v0. 8.0: A python toolbox to benchmark the robustness of machine learning models," *arXiv:1707.04131*, vol. 5, 2017.
[38] J. Lee Rodgers et al., "Thirteen ways to look at the correlation coefficient," *The American Statistician*, vol. 42, no. 1, pp. 59–66, 1988.
[39] Z. Wang et al., "Image quality assessment: from error visibility to structural similarity," *IEEE* ***TIP***, vol. 13, no. 4, pp. 600–612, 2004.