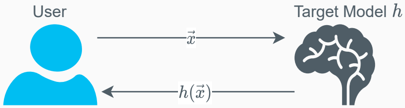
AMEBA: An Adaptive Approach to the Black-Box Evasion of Machine Learning Models

Stefano Calzavara **Lorenzo Cazzaro** Claudio Lucchese

Università Ca' Foscari Venezia

The rise of Machine Learning (ML)

Classification task

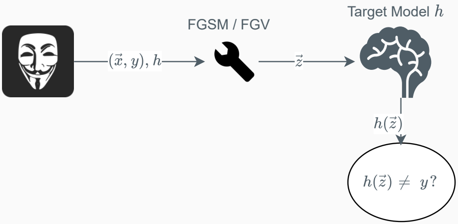


ML found a wide range of applications, in particular supervised learning.
Key service: classification

- A classifier (ML model) $h : \mathcal{X} \mapsto \mathcal{Y}$ is a function assigning a class label $y \in \mathcal{Y}$ to each element $\vec{x} \in \mathcal{X}$.
- Classifiers are normally trained on a training set, i.e., a set of correctly labeled instances $\{(\vec{x}_i, y_i))\}_i$.
- ML is vulnerable in an adversarial setting! The attacker is defined as $A : \mathcal{X} \rightarrow 2^{\mathcal{X}}$, that maps each instance into a set of possible perturbations.

Black-box evasion attacks

White-box evasion attack



Evasion attack

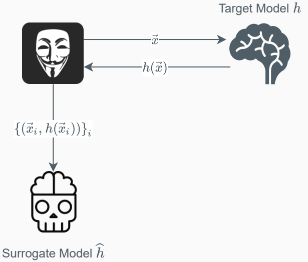
Given a classifier h and an instance $\vec{x}$ such that $h(\vec{x}) = y$, an *evasion attack* against $\vec{x}$ is any instance $\vec{z} \in A(\vec{x})$ such that $h(\vec{z}) \neq y$.

How to generate an evasion attack?

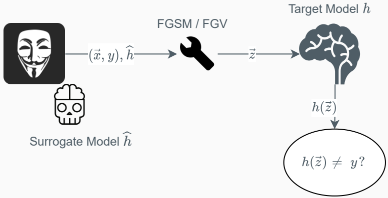
- In the white-box setting, the attacker has full knowledge of h and exploits methods like Fast Gradient Sign Method (FGSM)/ Fast Gradient Value (FGV).
- In the black-box setting, the attacker has no knowledge about h and limited access to it.

Two-step attack strategy

Step 1: Surrogate Model Training



Step 2: Evasion Attack Crafting



Transferability property

Evasion attacks often generalize across different ML models.

The attacker can adopt the two steps attack strategy [1]:

1. The attacker trains a surrogate model $\hat{h}$ using information extracted from h.

2. The attacker generates evasion attacks $\vec{z}$ against $\hat{h}$ and "transfers" them to h.

Two-step attack strategy - criticalities

Objective in the black-box setting

The attacker's budget, i.e., the number of queries to the target model, often is limited, e.g., query access might require a payment, like in the case of the Google Cloud Vision API.

Objective: Maximize the number of successful evasion attacks given a limited budget.

Two-step attack strategy - criticalities

Objective in the black-box setting

The attacker's budget, i.e., the number of queries to the target model, often is limited, e.g., query access might require a payment, like in the case of the Google Cloud Vision API.

Objective: Maximize the number of successful evasion attacks given a limited budget.

Inherent tension between the two steps

Two conflicting needs emerge:

- The attacker needs to query the target model to disclose its behavior.
- The attacker wants to query the target model with as many evasion attacks as possible.

Two-step attack strategy - criticalities

Objective in the black-box setting

The attacker's budget, i.e., the number of queries to the target model, often is limited, e.g., query access might require a payment, like in the case of the Google Cloud Vision API.

Objective: Maximize the number of successful evasion attacks given a limited budget.

Inherent tension between the two steps

Two conflicting needs emerge:

- The attacker needs to query the target model to disclose its behavior.
- The attacker wants to query the target model with as many evasion attacks as possible.

Disadvantages of the traditional two-steps attack

- The two steps are strictly separated.
- The strategy is sub-optimal.

Research questions

Why should the attacker follow a fixed two-step strategy and not resort to a more sophisticated approach which dynamically learns how to behave?

Research questions

Why should the attacker follow a fixed two-step strategy and not resort to a more sophisticated approach which dynamically learns how to behave?

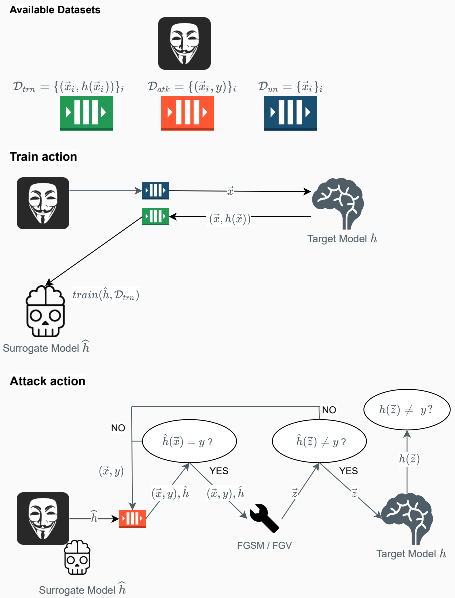
Solution

We present AMEBA, a new adaptive attack strategy, which dynamically learns whether queries to the target model should be leveraged for surrogate model training (step 1) or for evasion attack crafting (step 2).

In our paper:

1. Definition of the threat model.
2. Definition of AMEBA through the reduction from the two-steps evasion attack problem to the Multi-Armed Bandit (MAB) problem.
3. Experimental evaluation on public datasets and discussion.

Threat model



Available Datasets

The attacker has access to 3 datasets (queues):

- $\mathcal{D}_{trn}$ used for surrogate model training.
- $\mathcal{D}_{atk}$ used for evasion attacks crafting.
- $\mathcal{D}_{un}$ used to collect labels from h.

Available Actions

Train: the attacker asks h for a prediction and trains $\hat{h}$.

Attack: the attacker crafts $\vec{z}$ against $\hat{h}$ from $\vec{x} \in \mathcal{D}_{atk}$ and submits $\vec{z}$ to h, if possible. Otherwise the attacker pushes $(\vec{x}, y)$ in $\mathcal{D}_{atk}$.

Multi-Armed Bandit (MAB)

MAB optimization problem with Bernoulli-Beta bandits

Given a set of $K \geq 2$ possible actions $\mathcal{A} = \{a_1, ..., a_K\}$ and $T \geq 1$ rounds, MAB requires to choose the sequence of T actions from $\mathcal{A}$ which maximizes a reward.
The assumptions are:

- The rewards are 0 or 1 for each action and are distributed according to a Bernoulli probability distribution independent and different for each action.
- It is only possible to observe the reward for the selected action.
- θ_{a_k} is the unknown mean reward (probability of success) of action a_k.

A well-known solution is given by the Thompson Sampling algorithm [3].

Reduction and AMEBA

MAB	Two-steps evasion problem
Number of rounds T	Attacker's budget (one query per round)
Set of actions $\mathcal{A} = \{a_1, \ldots, a_k\}$	$\mathcal{A} = \{Train, Attack\}$
Rewards r_{a_k}	$r_{Train} = 1$ if $similarity(h, \hat{h})$ improves $r_{Attack} = 1$ if $h(\vec{z}) \neq y$

Reduction and AMEBA

MAB	Two-steps evasion problem
Number of rounds T	Attacker's budget (one query per round)
Set of actions $\mathcal{A} = \{a_1, \ldots, a_k\}$	$\mathcal{A} = \{Train, Attack\}$
Rewards r_{a_k}	$r_{Train} = 1$ if $similarity(h, \hat{h})$ improves $r_{Attack} = 1$ if $h(\vec{z}) \neq y$

Why is the rewards scheme effective?

- Low success rate $Attack \implies$ improve $similarity(h, \hat{h})$.
- $similarity(h, \hat{h})$ reaches a plateau $\implies$ exploit $Attack$.

Reduction and AMEBA

MAB	Two-steps evasion problem
Number of rounds T	Attacker's budget (one query per round)
Set of actions $\mathcal{A} = \{a_1, \ldots, a_k\}$	$\mathcal{A} = \{Train, Attack\}$
Rewards r_{a_k}	$r_{Train} = 1$ if $similarity(h, \hat{h})$ improves $r_{Attack} = 1$ if $h(\vec{z}) \neq y$

Why is the rewards scheme effective?

- Low success rate *Attack* $\implies$ improve $similarity(h, \hat{h})$.
- $similarity(h, \hat{h})$ reaches a plateau $\implies$ exploit *Attack*.

AMEBA implementation

- AMEBA can be simply defined using a MAB solving algorithm!
- What happens if *Attack* cannot be perfomed? Perform the *Train* action!
- $similarity(h, \hat{h}) = \text{CROSSVALSCORE}(\hat{h}, \mathcal{D}_{trn})$.

Experimental evaluation

We compare AMEBA against the traditional two-steps attack strategy in terms of:

- Number of successful evasion attacks.
- Transferability of the evasion attacks.

Experimental evaluation

We compare AMEBA against the traditional two-steps attack strategy in terms of:

- Number of successful evasion attacks.
- Transferability of the evasion attacks.

Evasion attacks are crafted using the FGV method [2].

Experimental datasets and settings:

Dataset	$\|\mathcal{D}_{trn}\|$	$\|\mathcal{D}_{atk}\|$	ϵ	attacker's budget	**Surrogate**	**Target**	**Target accuracy**
Spambase	100	900/ 1900	0.10/ 0.15	900/ 1900	Linear SVM	RandomForest	0.96
						AdaBoost	0.97
						Logistic Regression	0.93
Wine	100	900/ 1900	0.20/ 0.25	900/ 1900	Linear SVM	RandomForest	0.99
						AdaBoost	0.99
						Logistic Regression	0.99
CodRNA	100	900/ 1900	0.10/ 0.15	900/ 1900	Linear SVM	RandomForest	0.97
						AdaBoost	0.97
						Logistic Regression	0.95
MNIST	100	1900/ 2900	3	2900	LeNet	MODEL A	0.99
						MODEL A DROPLESS	0.99
						MODEL C	0.99
						CNN	0.99

Experimental results - AMEBA vs baseline

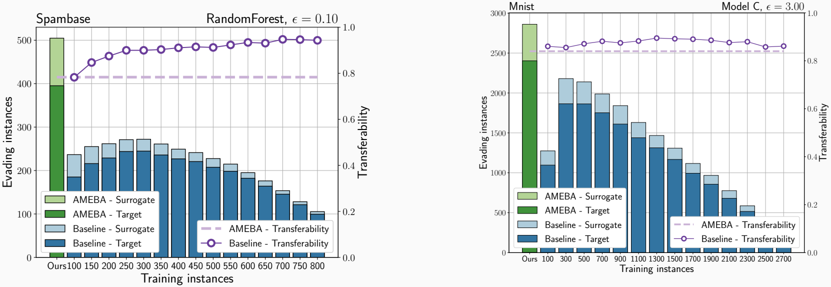


Figure: AMEBA VS two steps attack strategy.
On the left, results for Spambase dataset, $T = 1000$.
On the right, results for the MNIST dataset, $T = 1000$.

Across all datasets, perturbations and budgets, improvements on the number of successful evasion attacks range from 5% to 75%.

Discussion

Why does AMEBA work?

- AMEBA effectively alternates the two actions.
- Organize $\mathcal{D}_{atk}$ as queue is fundamental, since AMEBA dynamically refines the surrogate model. Then the remaining $\vec{x} \in \mathcal{D}_{atk}$ could be exploited effectively later.

Discussion

Why does AMEBA work?

- AMEBA effectively alternates the two actions.
- Organize $\mathcal{D}_{atk}$ as queue is fundamental, since AMEBA dynamically refines the surrogate model. Then the remaining $\vec{x} \in \mathcal{D}_{atk}$ could be exploited effectively later.

Performance

- An attacker can carry out an adaptive black-box attack just in a matter of minutes.
- The average time spent to craft a successful evasion attack is less than 2 seconds.

Conclusion

Defects of the traditional two-steps attack strategy

The traditional two steps-attack strategy in the black-box setting is sub-optimal.

Conclusion

Defects of the traditional two-steps attack strategy

The traditional two steps-attack strategy in the black-box setting is sub-optimal.

AMEBA

- AMEBA outperforms the traditional strategy since it infers how to alternate the *Train* and *Attack* actions.
- AMEBA effectively solves the delicate trade-off in the use of queries to maximize the number of successful evasion attacks.

Conclusion

Defects of the traditional two-steps attack strategy

The traditional two steps-attack strategy in the black-box setting is sub-optimal.

AMEBA

- AMEBA outperforms the traditional strategy since it infers how to alternate the *Train* and *Attack* actions.
- AMEBA effectively solves the delicate trade-off in the use of queries to maximize the number of successful evasion attacks.

Future works

- Experiment different rewards for the *Train* action.
- Generalize the approach to the case where the output is a confidence score.

References

[1] N. Papernot, P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami.
Practical black-box attacks against machine learning.
In R. Karri, O. Sinanoglu, A. Sadeghi, and X. Yi, editors, *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017*, pages 506–519. ACM, 2017.

[2] A. Rozsa, E. M. Rudd, and T. E. Boult.
Adversarial diversity and hard positive generation.
In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2016, Las Vegas, NV, USA, June 26 - July 1, 2016*, pages 410–417. IEEE Computer Society, 2016.

[3] A. Slivkins.
Introduction to multi-armed bandits.
Foundations and Trends in Machine Learning, 12(1-2):1–286, 2019.