The Robust Malware Detection Challenge and Greedy Random Accelerated Multi-Bit Search

AdvML & GRAMS

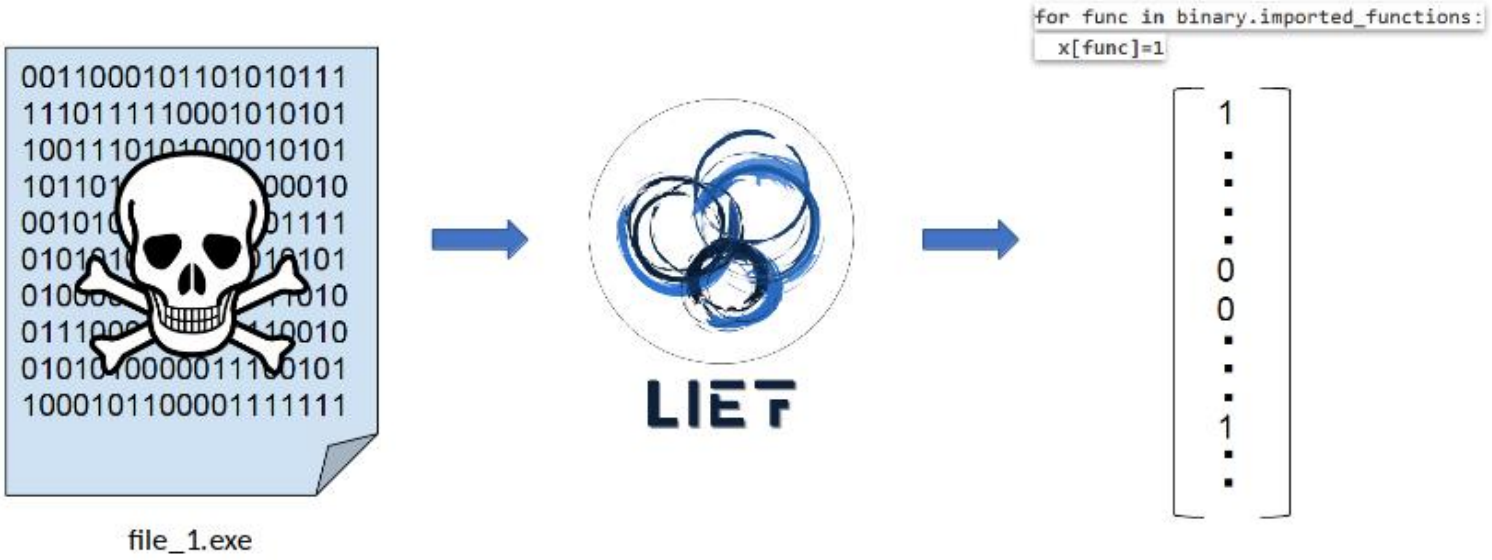
Sicco Verwer, Azqa Nadeem, Christian Hammerschmidt, Laurens Bliek, Abdullah Al-Dujaili, Una-May O'Reilly

Cyber Analytics Lab

TU Delft

The challenge

- Thanks to Abdullah Al-Dujaili and Una-May O'Reilly

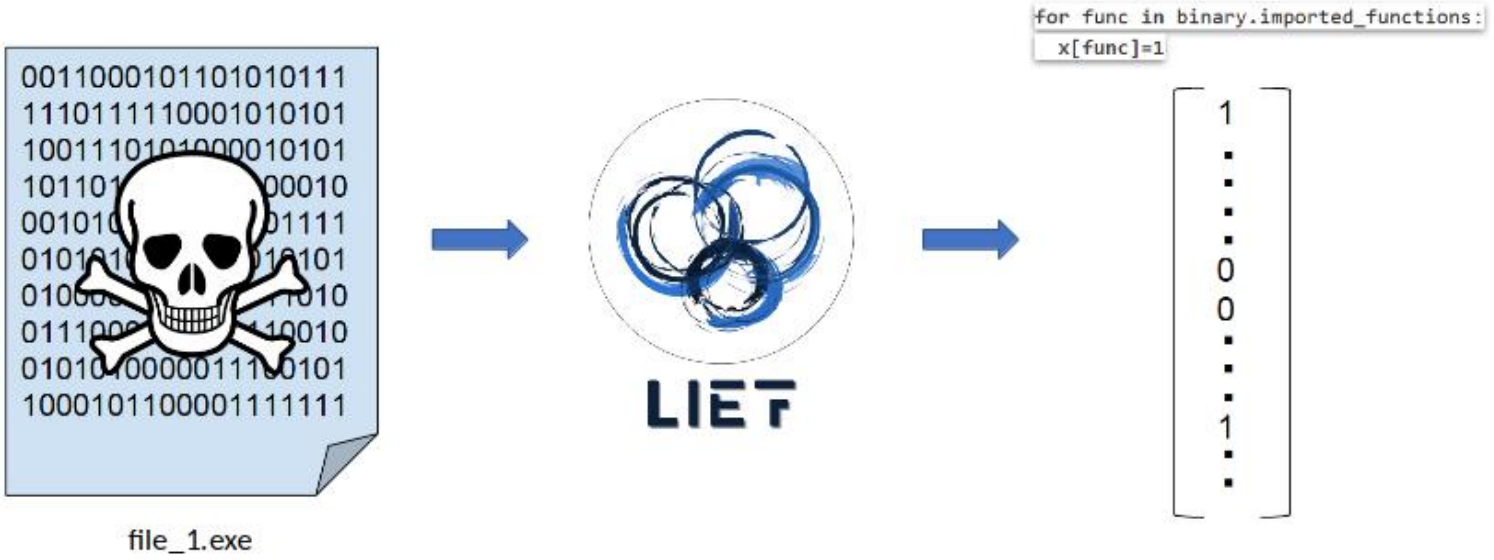


Cyber Analytics Lab

TU Delft

The challenge

- Thanks to Abdullah Al-Dujaili and Una-May O'Reilly

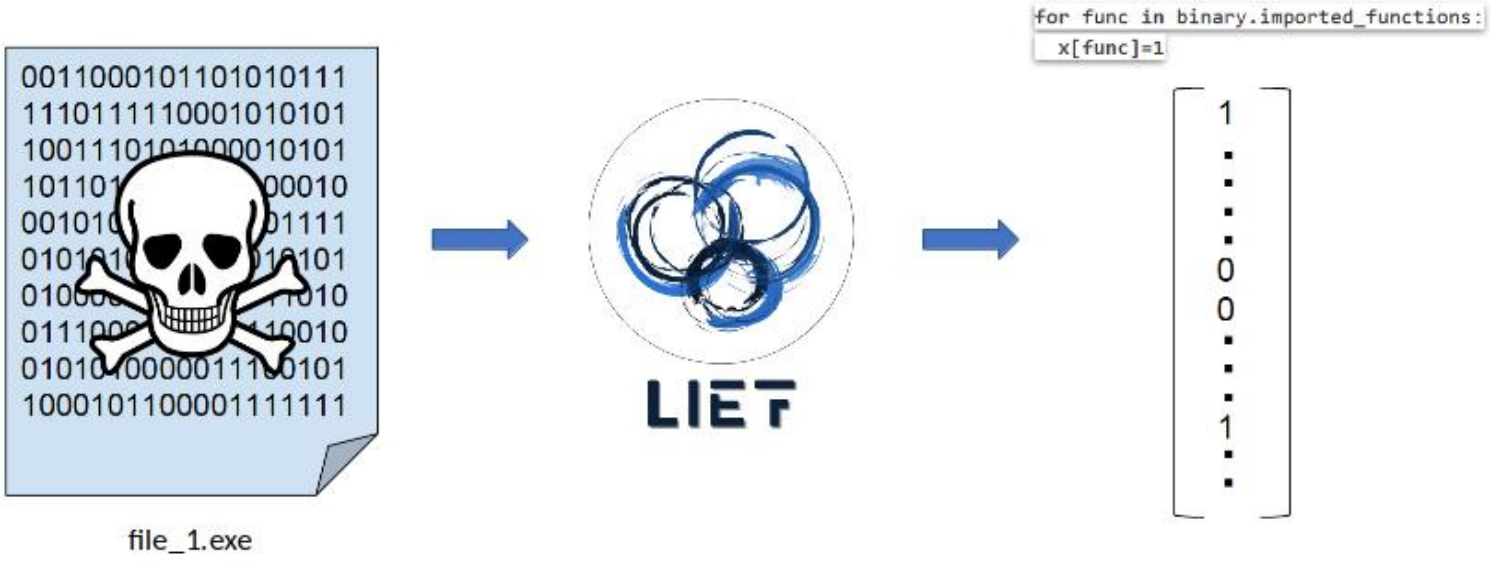


22.761 bits per file, 34.200 files
each bit indicates the presence of one sys call

Cyber Analytics Lab
TU Delft

The challenge

- Thanks to Abdullah Al-Dujaili and Una-May O'Reilly

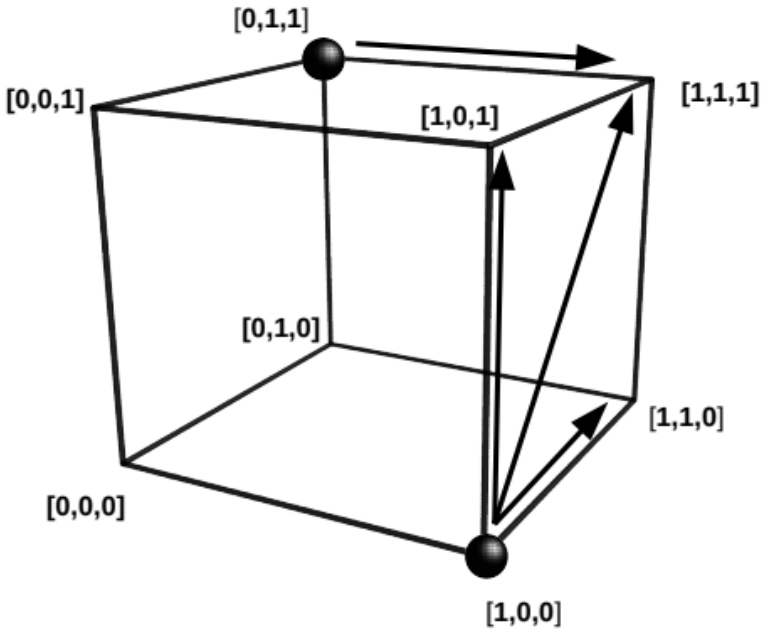


Classifiers obtain over 90% accuracy
Malware authors will evade, but how?

Cyber Analytics Lab
TU Delft

The threat model

- Problems in existing threat models:
  - An ε radius does not capture adding sys calls
  - Removing sys calls can nullify the malware
- Solution, or a step in the right direction:



Cyber Analytics Lab
TU Delft

The threat model

- Any number of 0 valued bits can be flipped to 1 values
- Use adversarial training to attack and defend

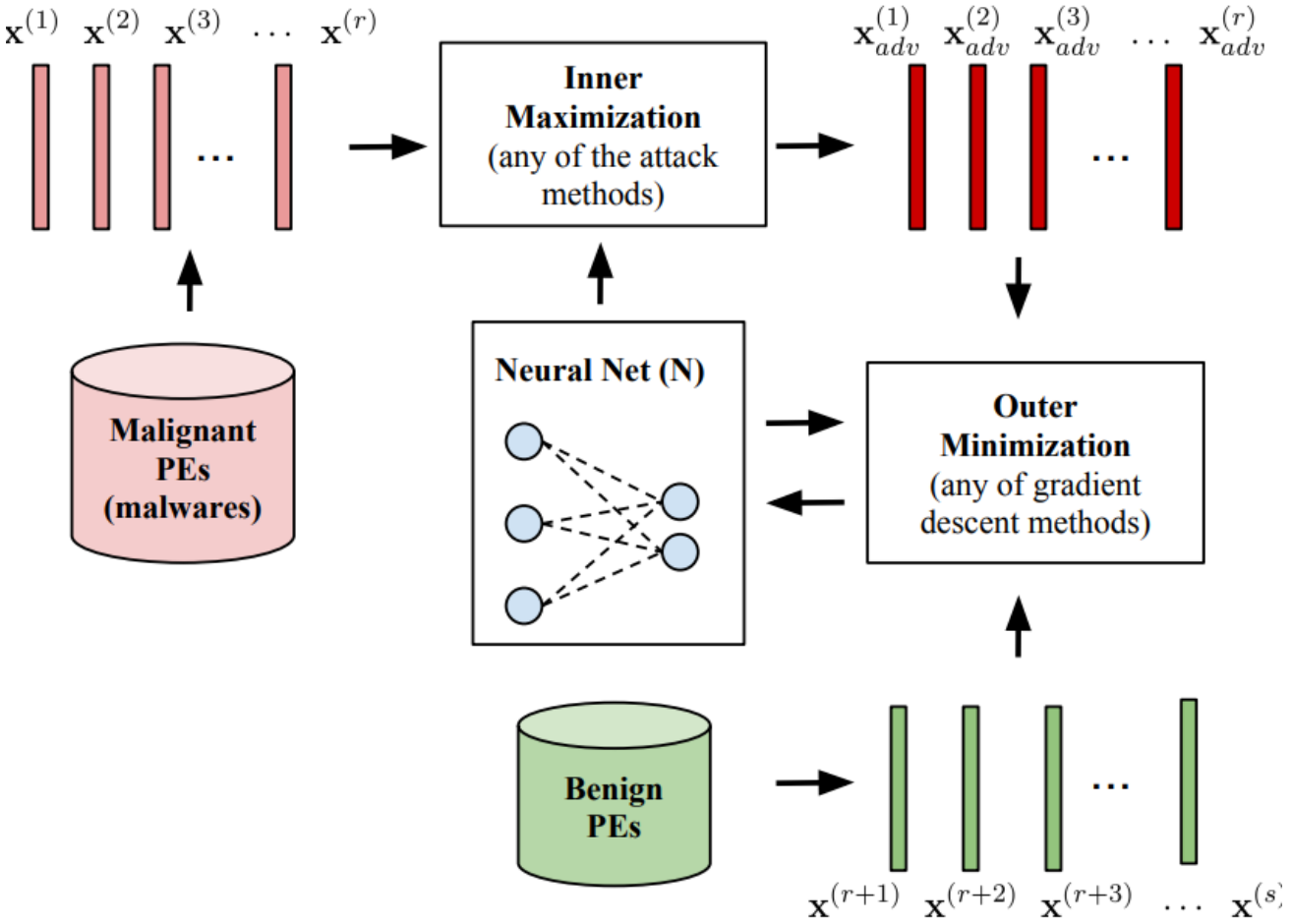
$$\theta^* \in \underbrace{\arg\min_{\theta \in \mathbb{R}^p} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left[\overbrace{\max_{\bar{\mathbf{x}} \in \mathcal{S}(\mathbf{x})} L(\theta, \bar{\mathbf{x}}, y)}^{\text{adversarial loss}} \right]}_{\text{adversarial learning}}$$

- Implemented in the SLEIPNIR framework

Al-Dujaili, A. Huang, E. Hemberg and U. O'Reilly, *Adversarial Deep Learning for Robust Detection of Binary Encoded Malware 2018 IEEE Security and Privacy Workshops (SPW)*
https://github.com/ALFA-group/robust-adv-malware-detection

Cyber Analytics Lab
TU Delft

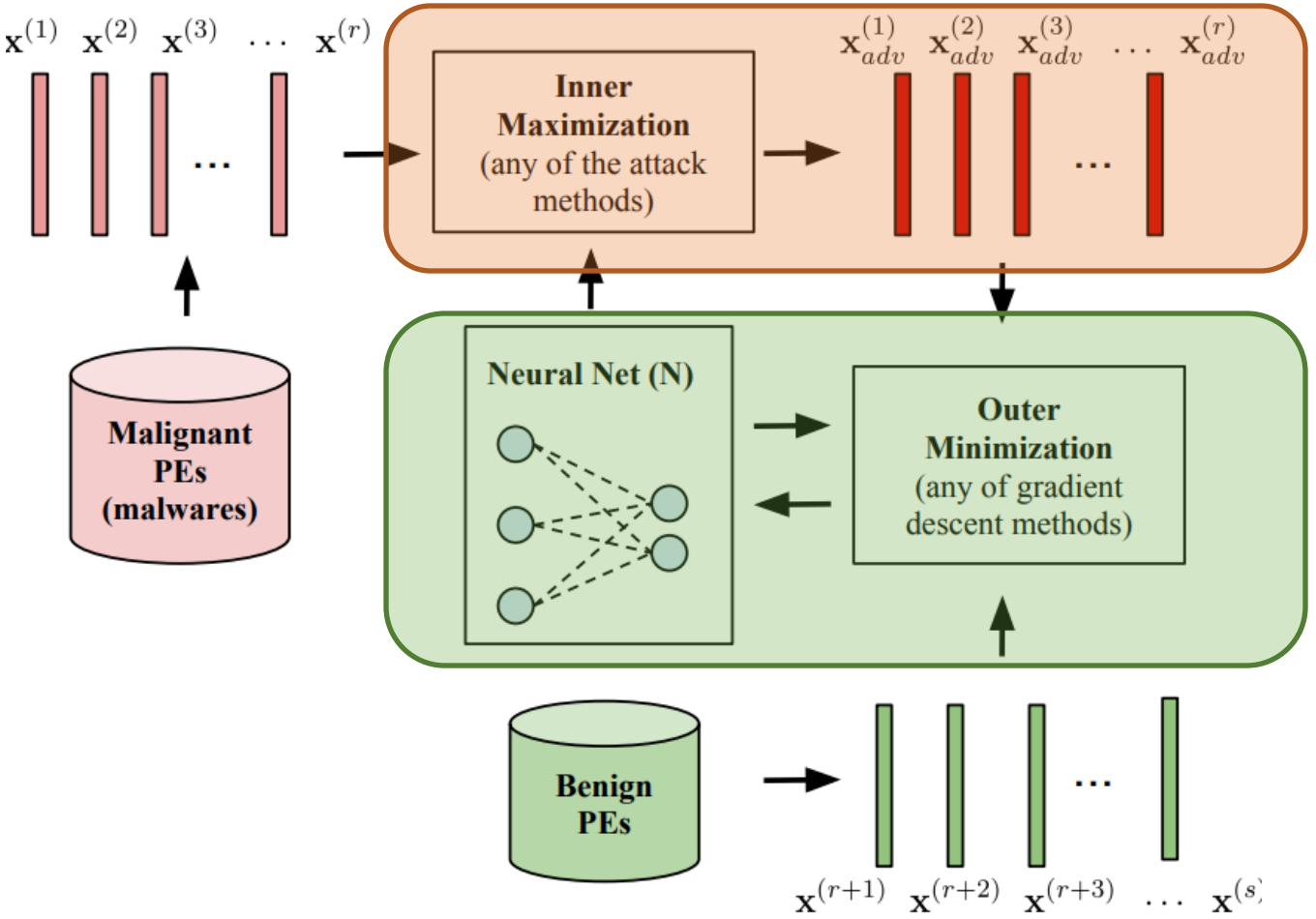
SLEIPNIR



Cyber Analytics Lab

TU Delft

Attack – inner maximizer



Defense – outer maximizer

Cyber Analytics Lab

TU Delft

Evasion rates

Model	Adversary				
	Natural	dFGSM^k	rFGSM^k	BGA^k	BCA^k
Natural	8.1	99.7	99.7	99.7	41.7
dFGSM^k	6.4	**6.4**	21.1	7.3	27.4
rFGSM^k	**5.7**	7.0	**5.9**	**5.9**	**6.8**
BGA^k	7.6	39.6	17.8	7.6	10.9
BCA^k	7.6	99.5	99.5	91.8	7.9

dFGSM^k = deterministic Fast Gradient Sign Method
BGA^k = Bitwise Gradient Ascend

Cyber Analytics Lab
TU Delft

The competition

Model	Adversary				
	Natural	dFGSM^k	rFGSM^k	BGA^k	?
Natural	8.1	99.7	99.7	99.7	?
dFGSM^k	6.4	**6.4**	21.1	7.3	?
rFGSM^k	**5.7**	7.0	**5.9**	**5.9**	
BGA^k	7.6	39.6	17.8	7.6	
BCA^k	7.6	99.5	99.5	91.8	7.9

Is there a better attacker?

Cyber Analytics Lab

TU Delft

The competition

Model	Adversary				
	Natural	dFGSM^k	rFGSM^k	BGA^k	?
Natural	8.1	99.7	99.7	99.7	?
dFGSM^k	6.4	**6.4**	21.1	7.3	?
rFGSM^k	**5.7**	7.0	**5.9**	**5.9**	
BGA^k	7.6	39.6	17.8	7.6	
?	?				7.9

Is there a better attacker?

Is there a better defender?

Cyber Analytics Lab

TU Delft

The competition

Model	Adversary				
	Natural	dFGSM^k	rFGSM^k	BGA^k	?
Natural	8.1	99.7	99.7	99.7	?
dFGSM^k	6.4	**6.4**	21.1	7.3	?
rFGSM^k	**5.7**	7.0	**5.9**	**5.9**	
BGA^k	7.6	39.6	17.8	7.6	
?	?				?

Is there a better attacker?

Is there a better defender?

What if they do not know each other?

Cyber Analytics Lab

TU Delft

The competition

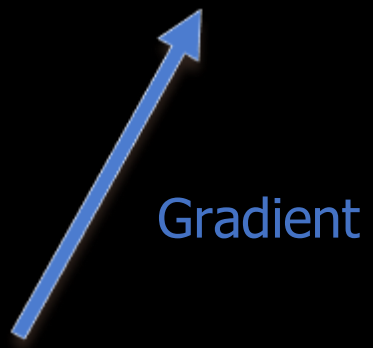
Model	Adversary				
	Natural	dFGSM^k	rFGSM^k	BGA^k	?
Natural	8.1	99.7	99.7	99.7	?
dFGSM^k	6.4	**6.4**	21.1	7.3	?
rFGSM^k	**5.7**	7.0	**5.9**	**5.9**	
BGA^k	7.6	39.6	17.8	7.6	
?	?				?

Is there a better attacker?

Our goal: build a new inner maximizer

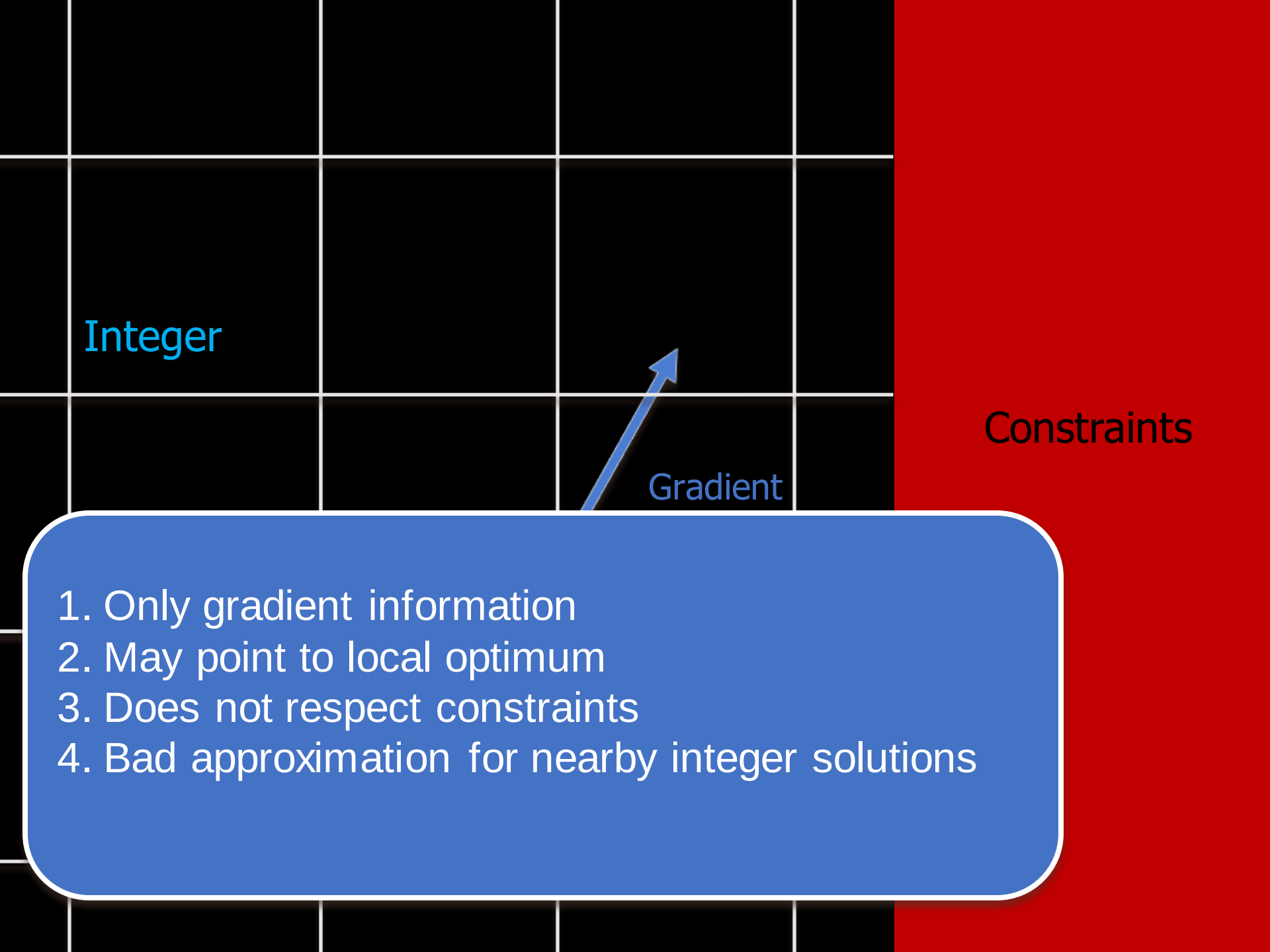
Cyber Analytics Lab

TU Delft

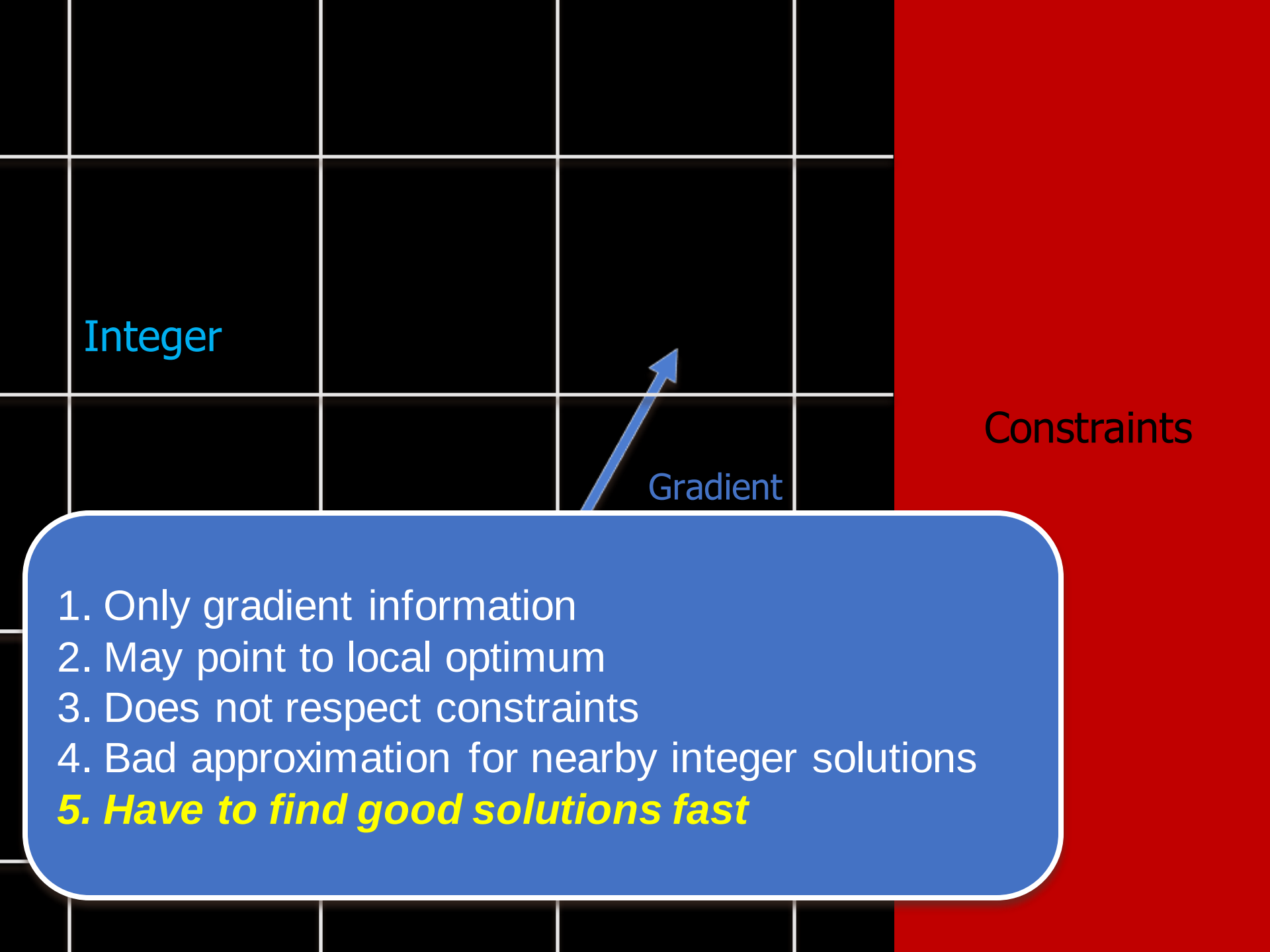

Gradient

Gradient

Constraints



1. Only gradient information
2. May point to local optimum
3. Does not respect constraints
4. Bad approximation for nearby integer solutions



1. Only gradient information
2. May point to local optimum
3. Does not respect constraints
4. Bad approximation for nearby integer solutions
5. ***Have to find good solutions fast***

GRAMS – an algorithm that

- Greedy – follows the gradient
- Random – is stochastic
- Adaptive – slow when difficult, fast when easy
- Multibit – flips multiple bits
- Search – …, well, it does hill-climbing

Cyber Analytics Lab

TUDelft

Greedy

- Given a batch of data points

0	0	1	1	0	1	0	0
0	1	0	0	1	0	1	1
0	0	1	0	1	1	1	1
0	1	0	1	1	0	0	0

- We perform bitflips, which ones to flip?

Greedy

- Given a batch of data points

0	0	1	1	0	1	0	0
0	1	0	0	1	0	1	1
0	0	1	0	1	1	1	1
0	1	0	1	1	0	0	0

- We perform bitflips, which ones to flip?
  - **Let's compute the gradient:**

0.5	-0.5	0.0	-1.3	1.5	-0.6	0.3	0.4
0.2	-0.9	0.1	0.1	1.2	0.4	-0.4	-0.2
-1.2	0.5	-0.3	0.7	-1.4	-0.2	0.4	0.2
0.8	-0.2	0.8	-0.3	0.2	0.5	-0.1	0.0

Greedy

- Given a batch of data points

0	0	1	1	0	1	0	0
0	1	0	0	1	0	1	1
0	0	1	0	1	1	1	1
0	1	0	1	1	0	0	0

- We perform bitflips, which ones to flip?
  - **Let's compute the gradient:**

0.5		0.0	-1.0	1.0	-0.6	0.3	0.4
0.2	-0.9	0.1	0.1		0.4	-0.4	-0.2
	0.5	-0.3	0.7	-1.0	-0.2		
0.8	-0.2	0.8	-0.3		0.5		0.0

keep to 0-1 domain

Cyber Analytics Lab

TU Delft

Greedy

- Given a batch of data points

0	0	1	1	0	1	0	0
0	1	0	0	1	0	1	1
0	0	1	0	1	1	1	1
0	1	0	1	1	0	0	0

- We perform bitflips, which ones to flip?
  - **Let's compute the gradient:**

0.5	0.0	1.0	0.3	0.4
0.2	0.1	0.1	0.4	
0.5		0.7		
0.8	0.8		0.5	0.0

only change 0 to 1 (wrt original point)

Cyber Analytics Lab

TU Delft

Greedy

- Given a batch of data points

0	0	1	1	0	1	0	0
0	1	0	0	1	0	1	1
0	0	1	0	1	1	1	1
0	1	0	1	1	0	0	0

- We perform bitflips, which ones to flip?
  - **Let's compute the gradient:**

0.5				1.0			
0.2						0.4	
	0.5			0.7			
0.8		0.8					

keep the top k in every row

Cyber Analytics Lab

TU Delft

Greedy

- Given a batch of data points

1	0	1	1	1	1	0	0
1	1	0	0	1	1	1	1
0	1	1	1	1	1	1	1
1	1	1	1	1	0	0	0

- We perform bitflips, which ones to flip?
  - **Let's compute the gradient:**

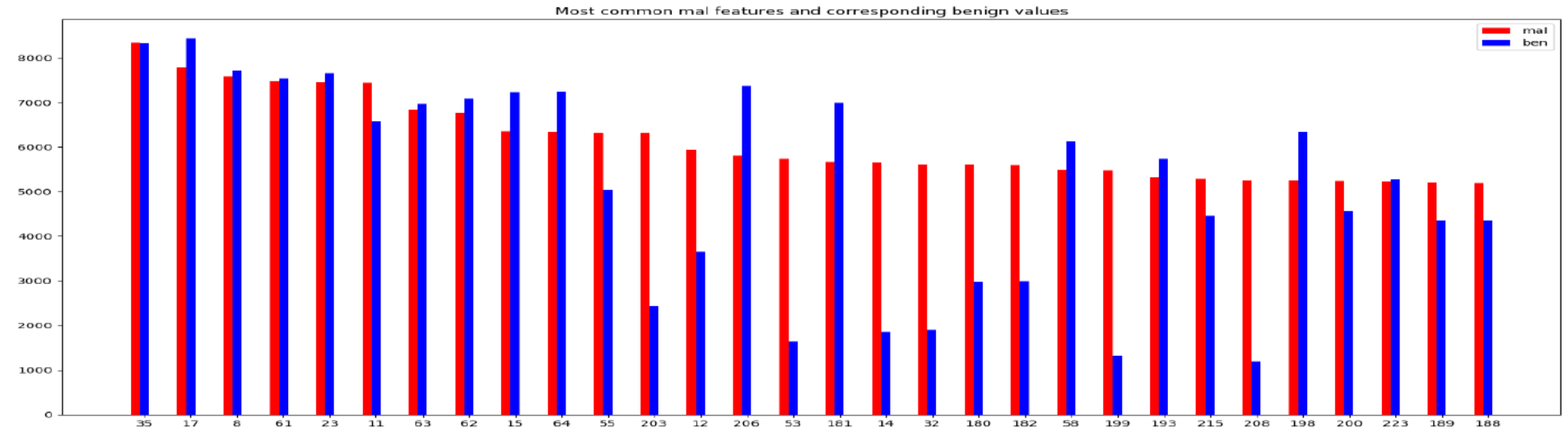
0.5				1.0			
0.2					0.4		
	0.5		0.7				
0.8		0.8					

flip these rows

Cyber Analytics Lab

TU Delft

Random



- But instead of fully random, fit a multivariate Bernoulli distribution *D* to the Benign datapoints
  - Sample bits from *D*, i.e., sample from Benign space
  - Restart 10 times
  - Follow gradients until convergence

Cyber Analytics Lab
TU Delft

Adaptive

- What value to use for *k*?
- Set *k* to 8
- If there is no improvement:
  - Divide *k* by 2
  - Undo the bit update
- If there in an improvement:
  - Multiply *k* by 2
- If *k < 0.5*
  - Stop

Results – attack – evasion rates

	Nat.	rFGSM	BGA	Grosse	GRAMS	AGA	GwT	ENA
Natural	6.9	**99.9**	**99.9**	99.8	93.9	94.2	**100.0**	84.7
rFGSM	5.8	5.8	5.8	**6.3**	**44.3**	0.0	0.0	27.6
AME-AT	5.8	5.8	5.8	**6.3**				
GRAMS	8.0	9.0	8.1	**9.7**				
NNWC	**10.8**	10.8	10.8	10.8				
NNNN	**5.4**	5.4	5.4	5.4				
RC	6.8	7.6	7.0	**8.5**				

?

GRAMS is the most effective attacker

Even knowing the defender, NN cannot be evaded

Cyber Analytics Lab

TU Delft

Results – attack – evasion rates

	Nat.	rFGSM	BGA	Grosse	GRAMS	AGA	GwT	ENA
Natural	6.9	**99.9**	**99.9**	99.8	93.9	94.2	**100.0**	84.7
rFGSM	5.8	5.8	5.8	**6.3**	**44.3**	0.0	0.0	27.6
AME-AT	5.8	5.8	5.8	**6.3**	**44.3**	0.0	0.0	29.6
GRAMS	8.0	9.0	8.1	**9.7**	**4.7**	0.0	0.0	2.6
NNWC	**10.8**	10.8	10.8	10.8	**3.7**	0.0	0.0	2.8
NNNN	**5.4**	5.4	5.4	5.4	**2.0**	0.0	0.0	1.6
RC	6.8	7.6	7.0	**8.5**	**27.2**	0.0	0.0	6.2

GRAMS is the most effective attacker against unknown defenses

Cyber Analytics Lab

TU Delft

Results – attack – evasion rates

	Nat.	rFGSM	BGA	Grosse	GRAMS	AGA	GwT	ENA
Natural	6.9	**99.9**	**99.9**	99.8	93.9	94.2	**100.0**	84.7
rFGSM	5.8	5.8	5.8	**6.3**	**44.3**	0.0	0.0	27.6
AME-AT	5.8	5.8	5.8	**6.3**	**44.3**	0.0	0.0	29.6
GRAMS	8.0	9.0	8.1	**9.7**	**4.7**	0.0	0.0	2.6
NNWC	**10.8**	10.8	10.8	10.8	**3.7**	0.0	0.0	2.8
NNNN	**5.4**	5.4	5.4	5.4	**2.0**	0.0	0.0	1.6
RC	6.8	7.6	7.0	**8.5**	**27.2**	0.0	0.0	6.2

GRAMS is the most effective attacker even against unknown defenses

but natural has a higher evasion rate!

Cyber Analytics Lab

TU Delft

Results – defense – F1 scores

	Nat.	rFGSM	BGA	Grosse	GRAMS	AGA	GwT	ENA
Natural	0.913	0.001	0.001	0.004	0.104	0.099	0.000	0.243
rFGSM	**0.921**	0.918	0.892	0.604	0.519	0.948	0.948	0.790
AME-AT	0.919	**0.919**	**0.919**	**0.917**	?			
GRAMS	0.905	0.899	0.904	0.895				
NNWC	0.880	0.880	0.880	0.880				
NNNN	0.883	0.883	0.883	0.883				
RC	0.918	0.914	0.917	0.909				

AME-AT obtains the best F1-scores against known attackers

Cyber Analytics Lab

TU Delft

Results – defense – F1 scores

	Nat.	rFGSM	BGA	Grosse	GRAMS	AGA	GwT	ENA
Natural	0.913	0.001	0.001	0.004	0.104	0.099	0.000	0.243
rFGSM	**0.921**	0.918	0.892	0.604	0.519	0.948	0.948	0.790
AME-AT	0.919	**0.919**	**0.919**	**0.917**	0.670	**0.949**	**0.949**	0.778
GRAMS	0.905	0.899	0.904	0.895	**0.922**	0.946	0.946	**0.933**
NNWC	0.880	0.880	0.880	0.880	0.917	0.936	0.936	0.922
NNNN	0.883	0.883	0.883	0.883	0.901	0.910	0.910	0.903
RC	0.918	0.914	0.917	0.909	0.797	0.953	0.953	0.921

AME-AT obtains the best F1-scores against known attackers
But GRAMS against unknown defenses!

Cyber Analytics Lab
TU Delft

Results – defense – F1 scores

	Nat.	rFGSM	BGA	Grosse	GRAMS	AGA	GwT	ENA
Natural	0.913	0.001	0.001	0.004	0.104	0.099	0.000	0.243
rFGSM	**0.921**	0.918	0.892	0.604	0.519	0.948	0.948	0.790
AME-AT	0.919	**0.919**	**0.919**	**0.917**	0.670	**0.949**	**0.949**	0.778
GRAMS	0.905	0.899	0.904	0.895	**0.922**	0.946	0.946	**0.933**
NNWC	0.880	0.880	0.880	0.880	0.917	0.936	0.936	0.922
NNNN	0.883	0.883	0.883	0.883	0.901	0.910	0.910	0.903
RC	0.918	0.914	0.917	0.909	0.797	0.953	0.953	0.921

AME-AT obtains the best F1-scores against known attackers
But GRAMS against unknown defenses!

but NNNN has the smallest evasion rates
therefore ties with GRAMS in the defense track

Cyber Analytics Lab
TU Delft

Conclusion

- SLEIPNIR AdvML framework
  - *Highlighy recommended for teaching AdvML*
- AdvML challenge:
  - Constrained and integer adversarial attack and defense
  - *Attackers do not know defenders*
  - *Defenders do not know attackers*
- GRAMS
  - Effective and efficient, a simple *multiple bit-flip search*
  - Open source and available at:
    - https://github.com/tudelft-cda-lab/GRAMS
- NNNN (not our work)
  - Cannot be attacked by flipping bits to 1