Sponge Examples: Energy-Latency Attacks on Neural Networks

Ilia Shumailov*^, Yiren Zhao*, Daniel Bates*, Nicolas Papernot^, Robert Mullins*, Ross Anderson*

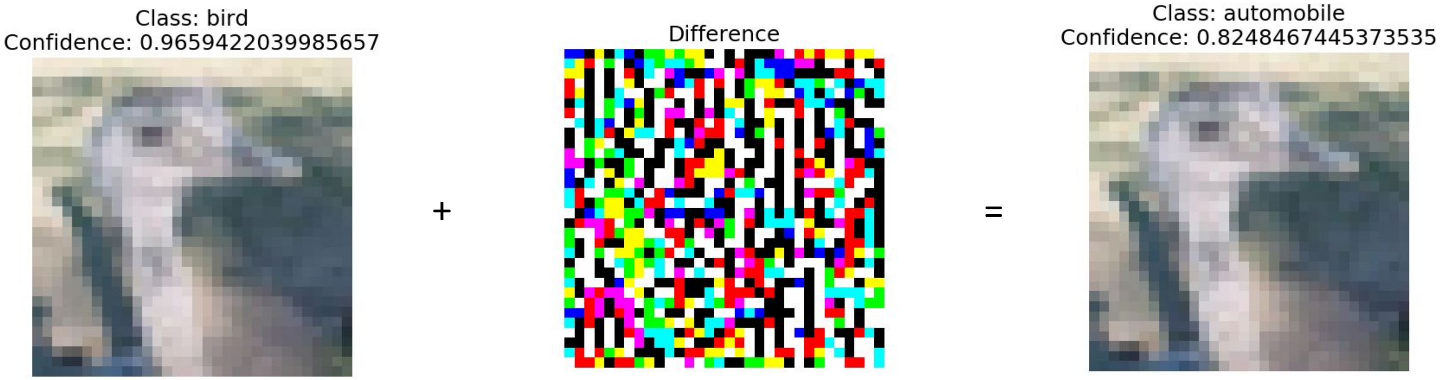
* University of Cambridge
^ University of Toronto, Vector Institute

Machine Learning

- Machine learning is everywhere
- We operate based on data, not formal rules
- There's a lot of non-determinism
- It is suddenly hard to define *Security*

https://xkcd.com/1838/

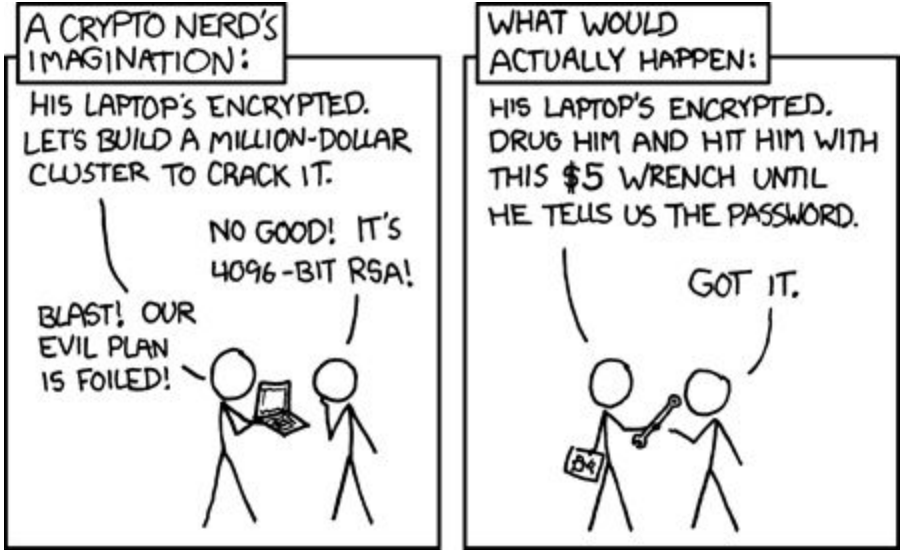
Computer Security in context of Machine Learning



- Adversarial examples exist for all models
- A large taxonomy of attackers
- Work in White / Grey / Black-box settings
- Attacks are scalable because of transferability

Machine Learning in context of Computer Security

- ML is a part of a larger pipeline
- As secure as the weakest component
- Clear threat model
- Safety and Security policies and cases
- Existence of trusted components
- Well defined environment



Machine Learning in context of Computer Security

r/MachineLearning · Posted by u/ProGamerGov 9 days ago

671

[D] Possible malware found hidden inside images from the ImageNet dataset

Discussion

I think I've discovered m
http://imagenet.stanfor

The following URLs show

- http://www. learnani
- http://www. pixelbird
- http://www. pixelbird

But when I posted my fi
find. I assumed this mea
Microsoft the files sayin
indeed malicious. The IP
numerous times in the p

Vulnerability Details : CVE-2018-8825

Google TensorFlow 1.7 and below is affected by: Buffer Overflow. The impact is: execute arbitrary code (local).

Publish Date : 2019-04-23 Last Update Date : 2019-04-25

Collapse All Expand All Select Select&Copy Scroll To Comments External Links

Search Twitter Search YouTube Search Google

– CVSS Scores & Vulnerability Types

CVSS Score	6.8
Confidentiality Impact	Partial (There is considerab
Integrity Impact	Partial (Modification of som attacker can affect is limite
Availability Impact	Partial (There is reduced pe
Access Complexity	Medium (The access conditi
Authentication	Not required (Authenticatio
Gained Access	None
Vulnerability Type(s)	Execute Code Overflow

TensorFlow models are programs

TensorFlow's runtime system interprets and executes programs.
programs that TensorFlow executes. TensorFlow programs are en
separately in **checkpoints**.

At runtime, TensorFlow executes the computation graph using th
may change depending on the parameters provided. TensorFlow
TensorFlow may read and write files, send and receive data over
performed with the permissions of the TensorFlow process. Allow

Machine Learning in context of Computer Security

system, hazard, risk, error, failure, threat, accident, safety case, security policy, trust, reliability, subject, person, principal, secrecy, privacy, confidentiality, anonymity, integrity, availability, authenticity, uncertainty, and *safety*

Machine Learning in context of Computer Security

system, hazard, risk, error, failure, threat, accident, safety case, security policy, trust, reliability, subject, person, principal, secrecy, privacy, confidentiality, anonymity, integrity, availability, authenticity, uncertainty, and *safety*

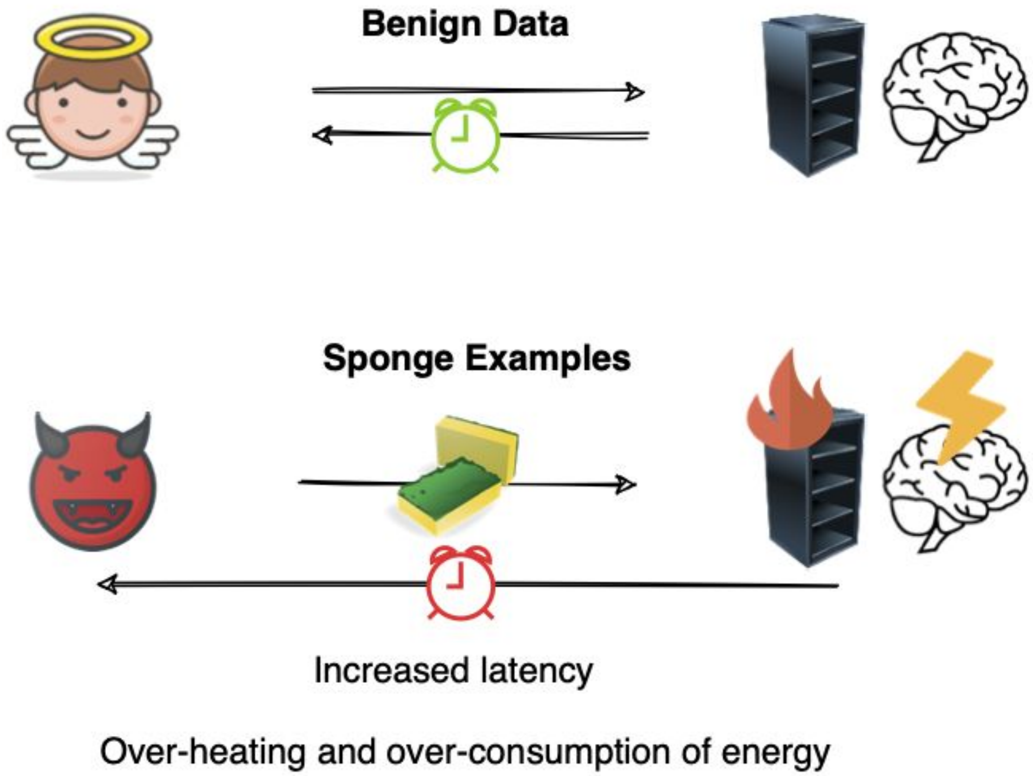
Safety looks at average case, **Security** considers worst case

What is a worst case for an ML component?

Availability

Ensuring **timely** and **reliable** access to and use of information. (NIST Special Publication 800-12)

Availability



Energy Gap

The amount of energy consumed by one inference pass (i.e. a forward pass in a neural network) depends primarily on:

- The overall **number of arithmetic operations** required to process the inputs;
- The **number of memory accesses** e.g. to the GPU DRAM.

Hypothesis 1: Data Sparsity

Optimisations exploit runtime **data sparsity** to increase efficiency.

- Zero-skipping multiplications;
- Encoding DRAM traffic to reduce the off-chip bandwidth requirement.

Hypothesis 2: Computation Dimensions

Modern networks have a **computational dimension**

- A large number of NLP models are **auto-regressive** e.g. RNNs and GPT2
- **Adaptive** input **dimensions** to help performance e.g. GPT2 uses Byte Pair Encoding
- ML components are **a part of loop**

Hypothesis 2: Computation Dimensions for GPT2

Auto-regressiveness adds an unbounded loop

Algorithm 1: Translation Transformer NLP pipeline

Result: y

$\downarrow O(l_{\text{tin}})$

$x_{\text{tin}} = \text{Tokenize}(x)$;

$y_{\text{touts}} = \emptyset$;

$\downarrow O(l_{\text{ein}})$

$x_{\text{ein}} = \text{Encode}\ (x_{\text{tin}})$;

$\downarrow O(l_{\text{tin}} \times l_{\text{ein}} \times l_{\text{tout}} \times l_{\text{eout}})$

while y_{tout} *has no end of sentence token* **do**

  $\downarrow O(l_{\text{eout}})$

  $y_{\text{eout}} = \text{Encode}\ (y_{\text{tout}})$;

  $\downarrow O(l_{\text{ein}} \times l_{\text{eout}})$

  $y_{\text{eout}} = \text{model.Inference}(x_{\text{ein}}, y_{\text{eout}}, y_{\text{touts}})$;

  $\downarrow O(l_{\text{eout}})$;

  $y_{\text{tout}} = \text{Decode}(y_{\text{eout}})$;

  $y_{\text{touts}}.\text{add}(y_{\text{tout}})$;

end

$\downarrow O(l_{\text{tout}})$;

$y = \text{Detokenize}(y_{\text{touts}})$

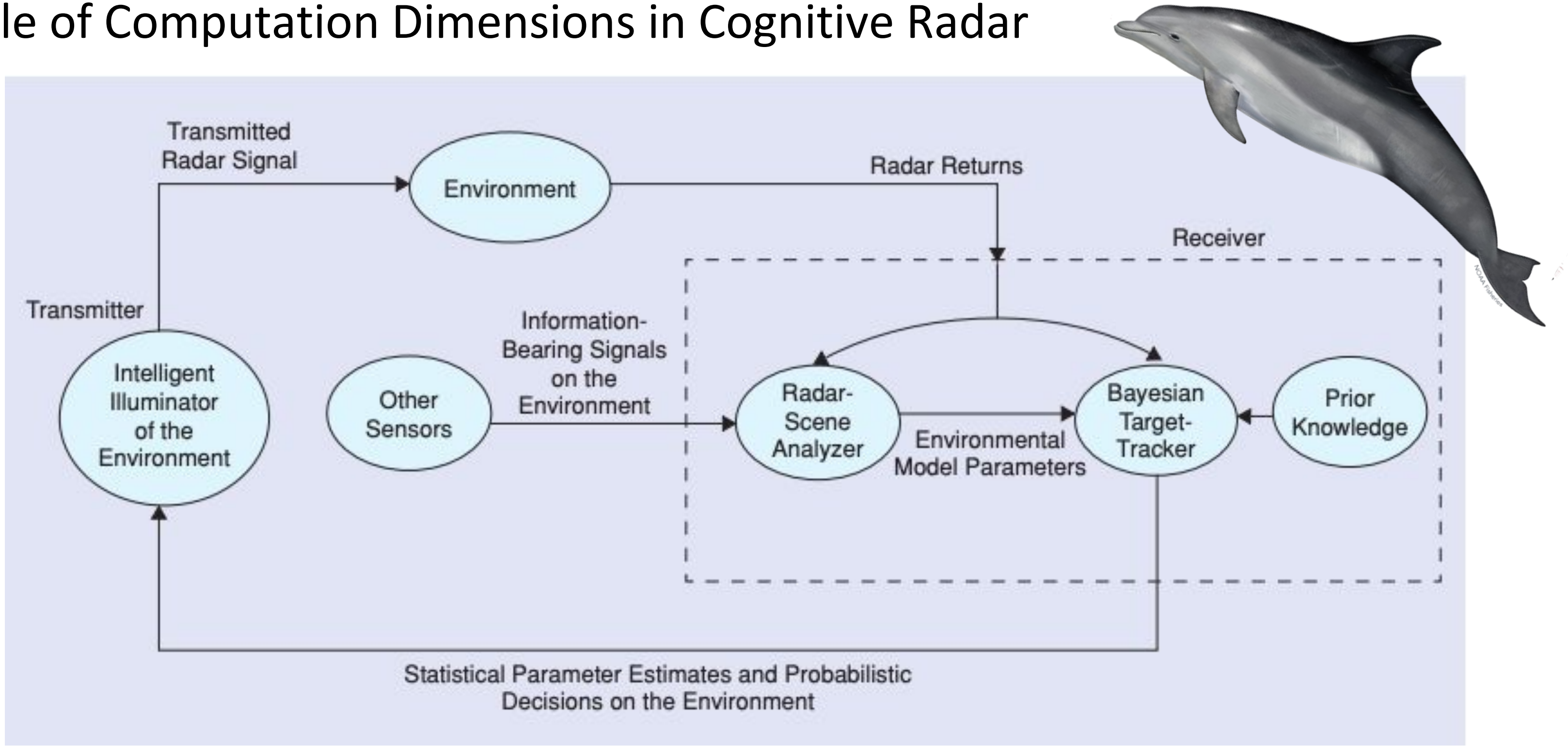
Encoding adds **variable** I/O representation

Benign with 4 tokens for input of size 16:
Athazagoraphobia => ath, az, agor, aphobia

1 error with 7 tokens for input of size 16:
Athazagoraphpbia => ath, az, agor, aph, p, bi, a

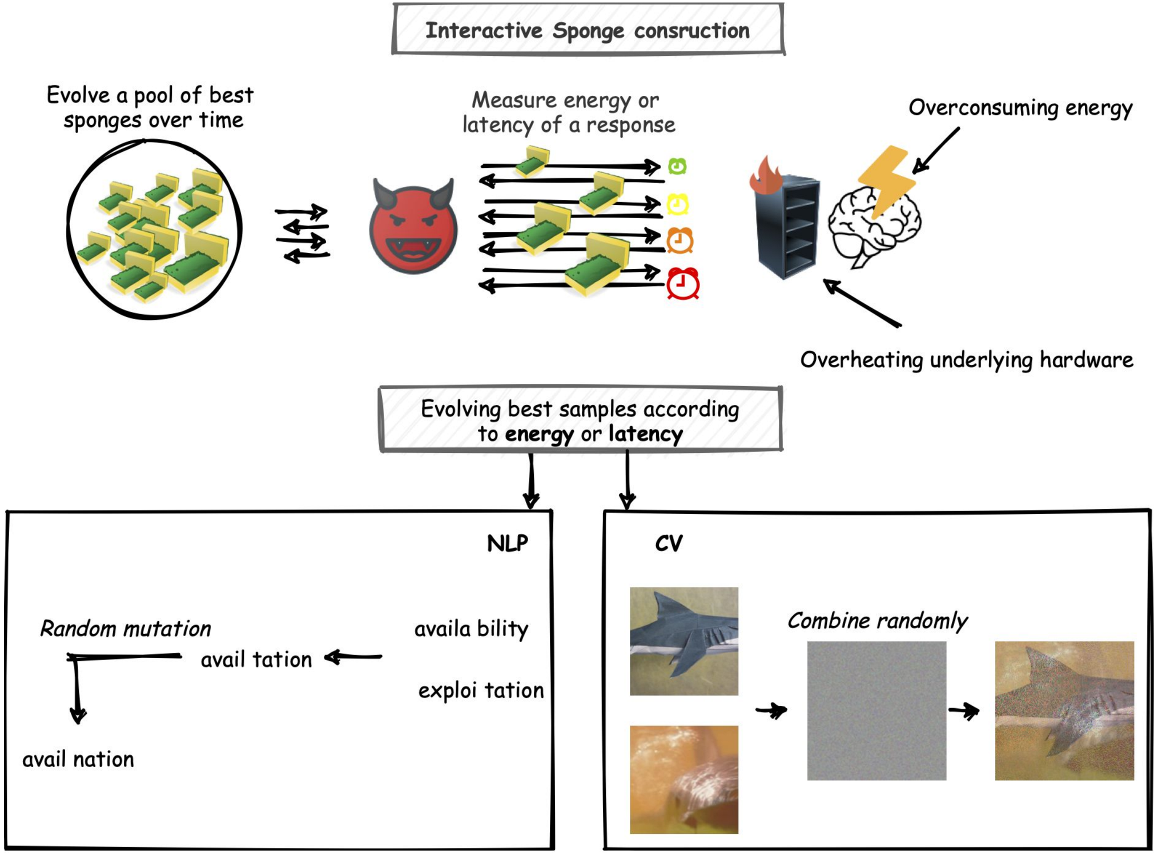
Malicious with 16 tokens for input of size 16:
A/h/z/g/r/p/p/i/ => A, /, h, /, z, /, g, /, r, /, p, /, p, /, i, /

Example of Computation Dimensions in Cognitive Radar



Block diagram of cognitive radar viewed as a dynamic closed-loop feedback system from *Cognitive radar: a way of the future*, Simon Haykin (2006)

Multiple ways to search for Sponge examples



Multiple ways to search for Sponge examples

- **White-box** gradient-based $-\sum_{a_l \in A} \|a_l\|_2$ i.e. large activation norms across all hidden layers
- **Interactive White-box**, **Grey-box** and **Black-box** genetic algorithm-based
  - Perform inference on a sample
  - Measure energy consumed or inference time
  - Combine worst performing samples
  - Mutate
  - Repeat
- **Blind Black-box** attack genetic algorithm-based
  - Pick model solving similar task or using similar dictionary
  - Perform transferability attack

White-box attack performance with NLP benchmarks

	Input size	$NVML_{gpu}$	$Natural_{asic}$	$Random_{asic}$	Sponge $Mean_{asic}$	Sponge Top $10\%_{asic}$	$Energy_{gpu}$	$Time_{gpu}$
Language Understanding: SuperGLUE Benchmark with [37]								
CoLA	15	5829.32	4.30	69.72	83.92	87.11	×20.25	×1.23
	30	9388.40	4.30	138.07	164.07	169.91	×39.51	×1.48
	100	22698.87	4.30	452.49	518.19	530.80	**×123.42**	**×3.82**
MNLI	15	6126.65	12.88	73.47	86.97	89.96	×6.98	×1.32
	30	9631.68	17.66	142.63	168.96	174.34	×9.87	×2.03
	100	22952.14	34.47	456.11	518.89	531.40	**×15.42**	**×3.16**
WSC	15	27876.53	14.48	523.28	1300.19	2152.67	×148.62	×9.83
	30	82822.58	34.94	1882.63	3927.63	5348.06	×153.08	×19.25
	100	662811.96	194.89	16754.13	25367.30	30692.95	**×157.49**	**×69.83**
Machine Translation: WMT14/16 with [41]								
En→Fr	30	59597.32	31.87	109.80	118.47	141.27	×4.43	×4.45
	50	93731.34	48.54	166.13	249.89	569.85	×11.74	×13.51
En→De	15	18133.66	18.19	35.80	242.39	542.35	×29.82	×32.86

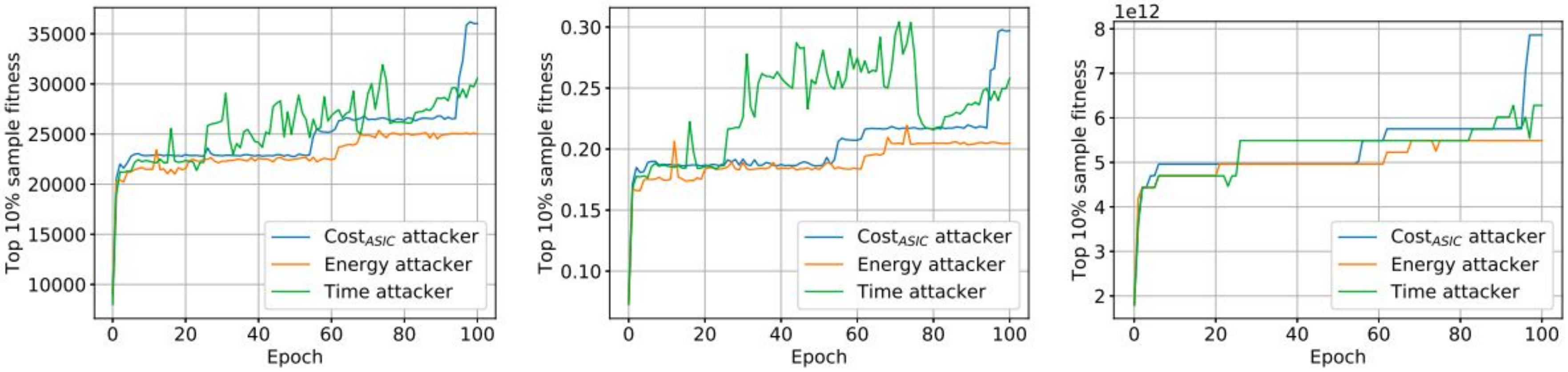
Energy is reported in millijoules. GA was ran for 100 epochs with a pool size of 100.

White-box attack performance for CV tasks

		$Time_{gpu}$ [s]	$Cost_{asic}$ [mJ]	$Cost_{asic}$ ratio	post-ReLU Density	Density	Max Density
ResNet-50	L-BFGS-B Sponge	0.011	164.727	0.863	0.619	0.885	0.998
	Sponge	0.016	160.887	0.843	0.562	0.868	
	Natural	0.017	160.562	0.842	0.572	0.867	
	Random	0.017	155.820	0.817	0.483	0.845	
DenseNet-121	L-BFGS-B Sponge	0.033	152.595	0.783	0.571	0.826	0.829
	Sponge	0.029	149.564	0.767	0.540	0.814	
	Natural	0.033	147.227	0.755	0.523	0.804	
	Random	0.030	144.365	0.741	0.487	0.792	
MobileNet v2	L-BFGS-B Sponge	0.011	87.511	0.844	0.692	0.890	0.996
	Sponge	0.010	84.513	0.815	0.645	0.868	
	Natural	0.011	85.075	0.821	0.646	0.873	
	Random	0.011	80.805	0.779	0.567	0.844	

Energy is reported in millijoules. GA was ran for 100 epochs with a pool size of 100.

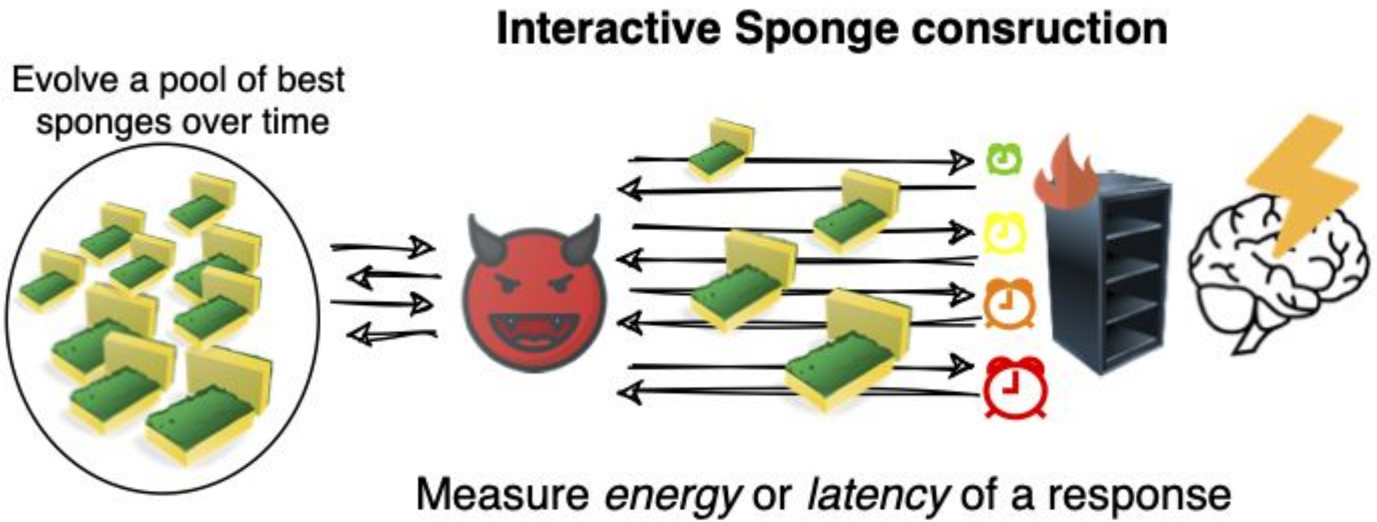
Interactive Black-box attack performance against WMT16 En→Fr benchmark



(a) Energy (b) Time (c) ASIC

Figure 1: Performance of Sponge Examples based on the Energy, Time and Simulator fitness costs.

Do sponges exist in practice? Yup



Towards defences again Sponge Examples

- Lesson from Computer security: **optimisations increase attack surface**
  - Side channel attacks
  - Denial-of-service attacks

- Optimisations **widen** average to worst case **time-energy gap**

- Not clear how to keep performance and security
  - Still have not solved Spectre & Meltdown
  - Constant time computation solves security issues, but things get too slow

- **Potential simple defense:**
  - Kill inference when more than average amount of time or energy is consumed
  - Will cause a lot of false positives and make jamming easy. Can we do better?

- **Real-time systems** in presence of Sponges
  - Can Tesla collision avoidance system afford to not make a decision?
  - What should be the maximum energy gap for RT?

Conclusions

- It is possible to attack model **availability** in both White and Black-box settings
- Attack can target **hardware optimisations**
  - For some CV tasks we fully negated benefits from acceleration
- Attacks can target **algorithmic complexity**
  - For some NLP tasks we managed to get up to **x200** energy consumption and **x70** time
- Average case is very different from worst case scenario
- **Impact** of ML **on climate change** might have been **underestimated**
- It is **not clear how to defend** systems against Sponge examples
- **Real-time systems** with ML components **should model availability** adversary

Thank you very much for listening!

Please do not hesitate to reach out in case there are any questions at

ilia.shumailov@cl.cam.ac.uk

https://arxiv.org/abs/2006.03463