AI and Cybersecurity: A Perfect Couple. . . or not?

Stjepan Picek

CROSSING 2023, September 18, 2023

Outline

Outline

Artificial Intelligence

- AI is the new electricity. (Andrew Ng)

- Computer vision.
- Healthcare.
- Natural Language Processing.
- Robotics.
- **Cybersecurity**.
- . . .

Artificial Intelligence

- Powerful hardware.
- Big data.
- Novel applications.

Artificial Intelligence

- Machine Learning → Deep Learning → Deep Neural Networks → . . . CNN, . . ., Transformers.
- Evolutionary algorithms.
- Almost anything you can think of and some things that should have never been considered.

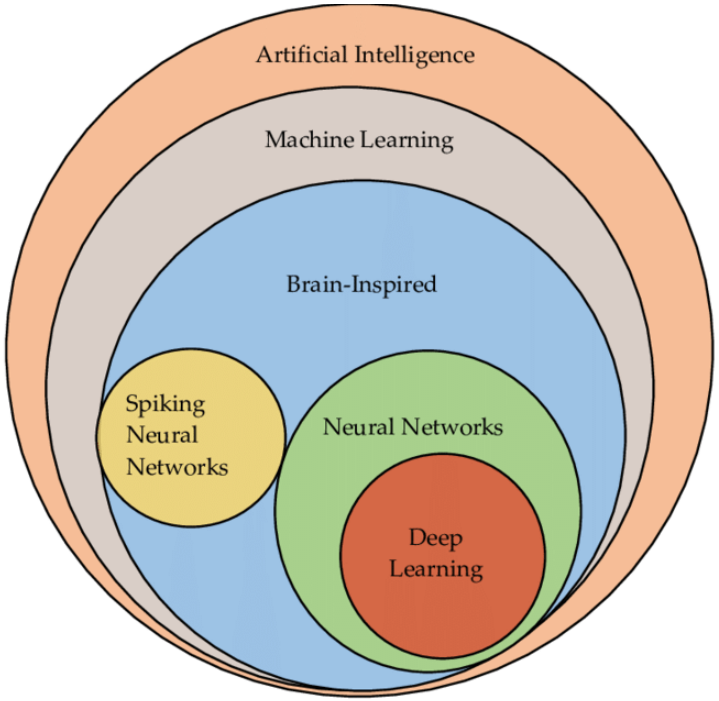


Figure: The Taxonomy of AI (Alom et al., 2019)

AI and Cybersecurity

- Artificial intelligence (AI) techniques are used more often in **attacks** than in **designs**.
- There are two main reasons for this:
  1. It is easier to validate that the attack works. Indeed, we require only a successful attack as proof. For designs, capturing all the notions of security when using data or fitness functions is difficult.
  2. Attacks are made after the designs are done. So, there is the effect of timeliness. For designs, one needs to use AI while constructing the system, which is often not possible. Later, even if AI produces improved constructions, it is hard to change the already-made design.

AI and Cybersecurity

- How to solve hard problems in cybersecurity?
- Problems need to be hard (to be worthwhile) but not too difficult (to be impossible to solve).
- Plenitude of problems and possible methods to solve them.
- Bruce Schneier CRYPTO-GRAM.

AI in Cybersecurity

- RNG and PRNG
- Side-channel and fault injection attacks
- Hardware Trojans
- Modelling attacks on PUFs
- Design of cryptographic primitives
- Cryptanalysis
- Intrusion detection
- Malware and spam identification and detection
- Adversarial machine learning
- Fuzzing
- Privacy-preserving machine learning
- . . .

Outline

Fault Injection

- A fault injection (FI) attack is successful if, after exposing the device to a specially crafted external interference, it shows an unexpected behavior exploitable by the attacker.
- FI can be divided into the characterization phase (finding faults) and using those faults for a successful attack.
- Insertion of signals has to be precisely tuned for the fault injection to succeed.
- Finding the correct parameters for a successful FI can be considered a search problem where one aims to find, within a minimum time, the parameter configurations that result in a successful fault injection.

Fault Injection

- Depending on the source of the fault, the search space of possible parameters changes significantly.
- Generally, the search space is too big to conduct an exhaustive search.
- Commonly, one defines several possible classes for classifying a single measurement:
  1. NORMAL: smart card behaves as expected, and the glitch is ignored
  2. RESET: smart card resets as a result of the glitch
  3. CHANGING: the response is changing when repeating measurements.
  4. SUCCESS: smart card response is a specific, predetermined value that does not happen under normal operation

Fault Injection

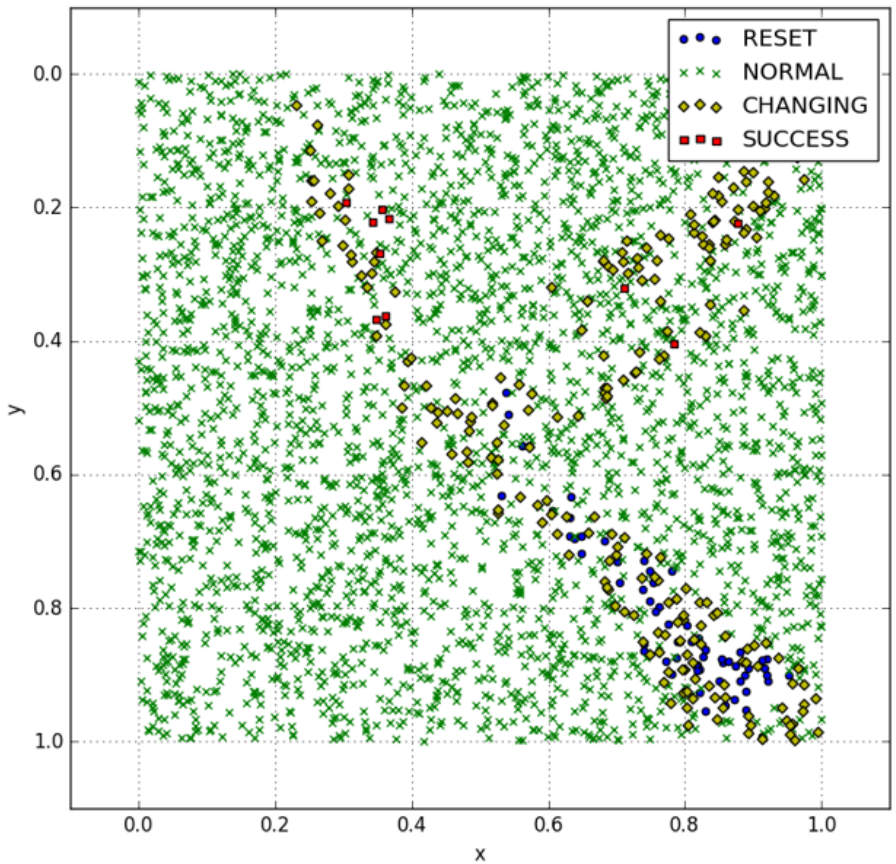
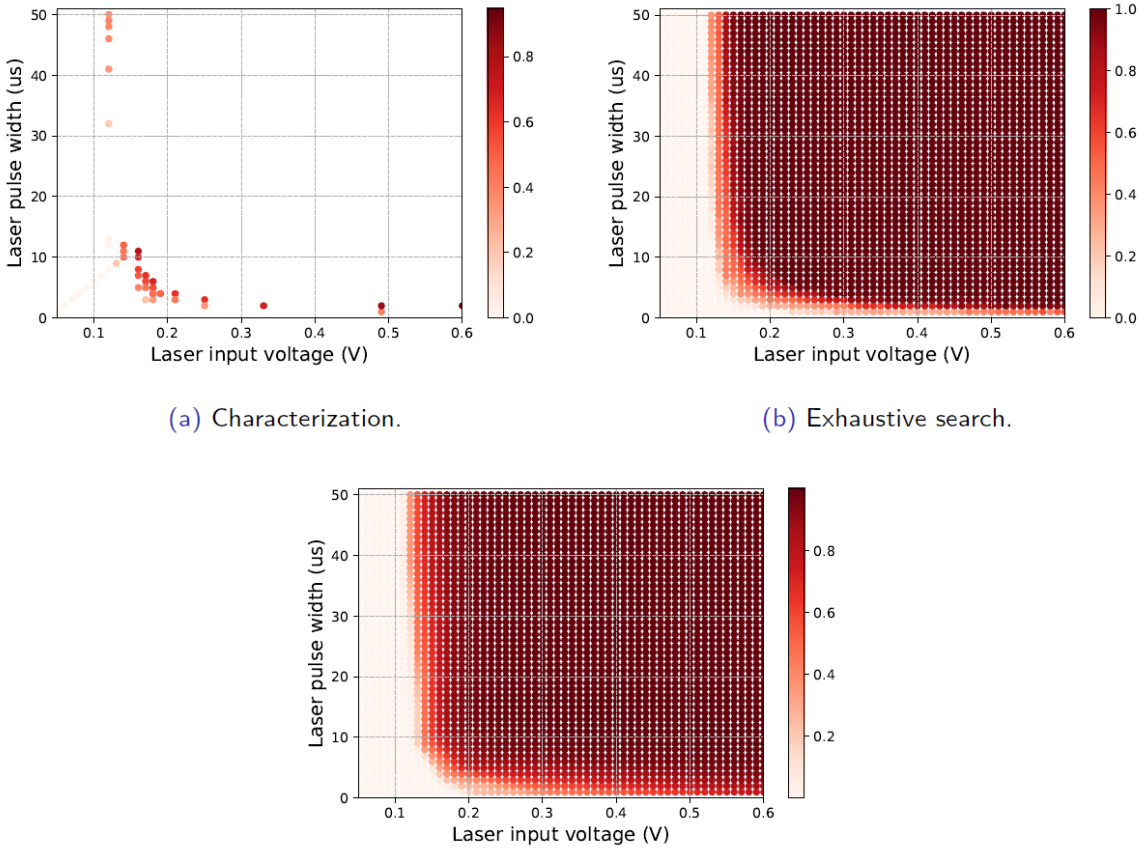


Figure: A depiction of search space for voltage glitching and two parameters.

Fault Injection



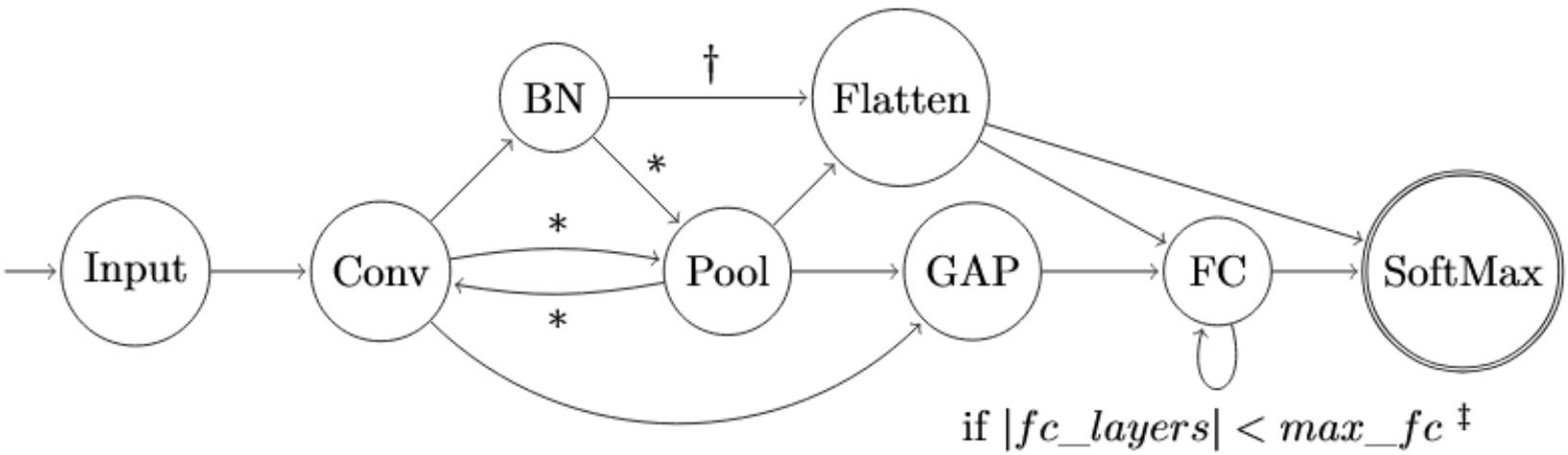
(a) Characterization.

(b) Exhaustive search.

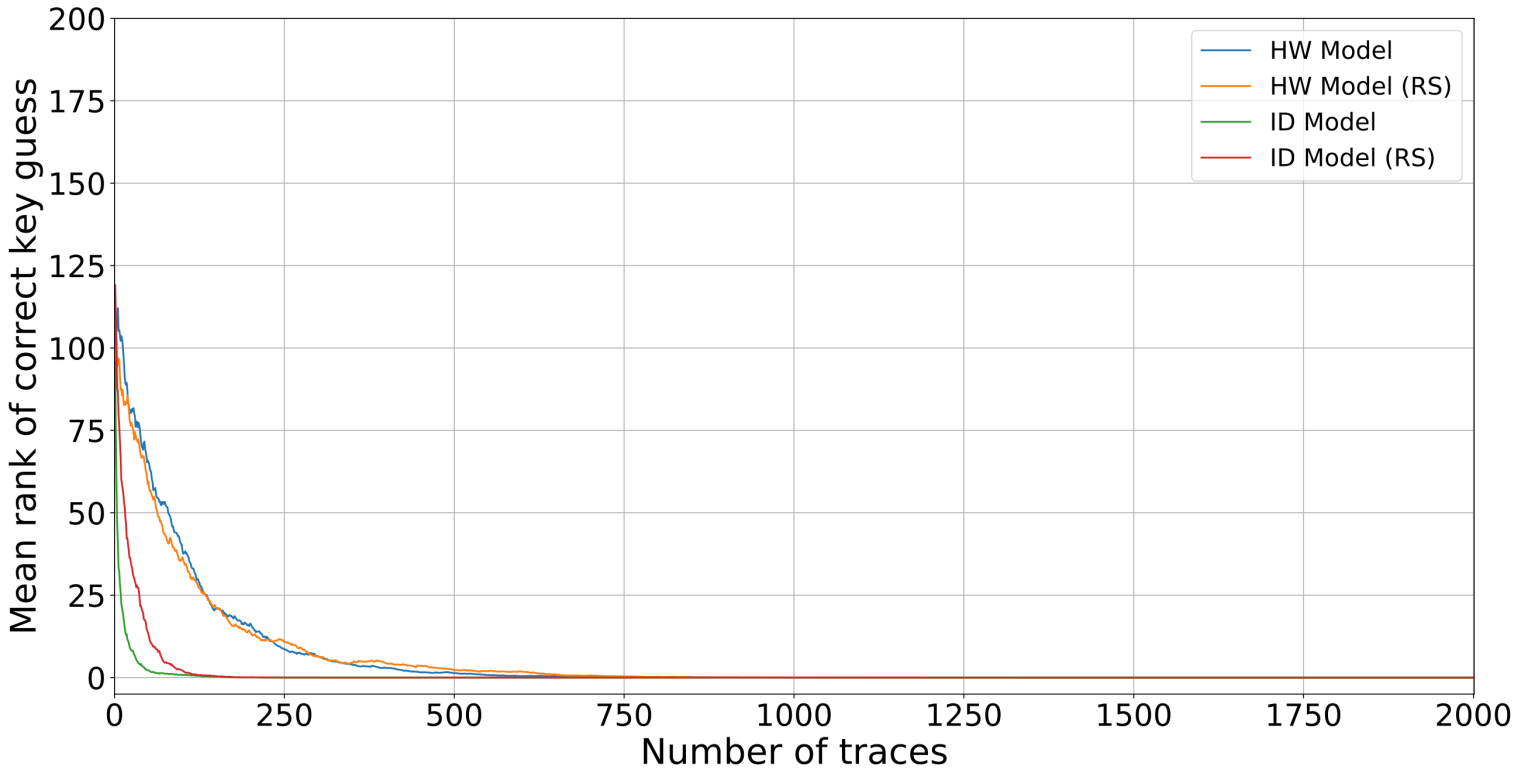
Figure: Deep learning prediction.

Reinforcement Learning

- Reinforcement learning attempts to teach an agent how to perform a task by letting the agent experiment and experience the environment, maximizing some reward signal.



Reinforcement Learning



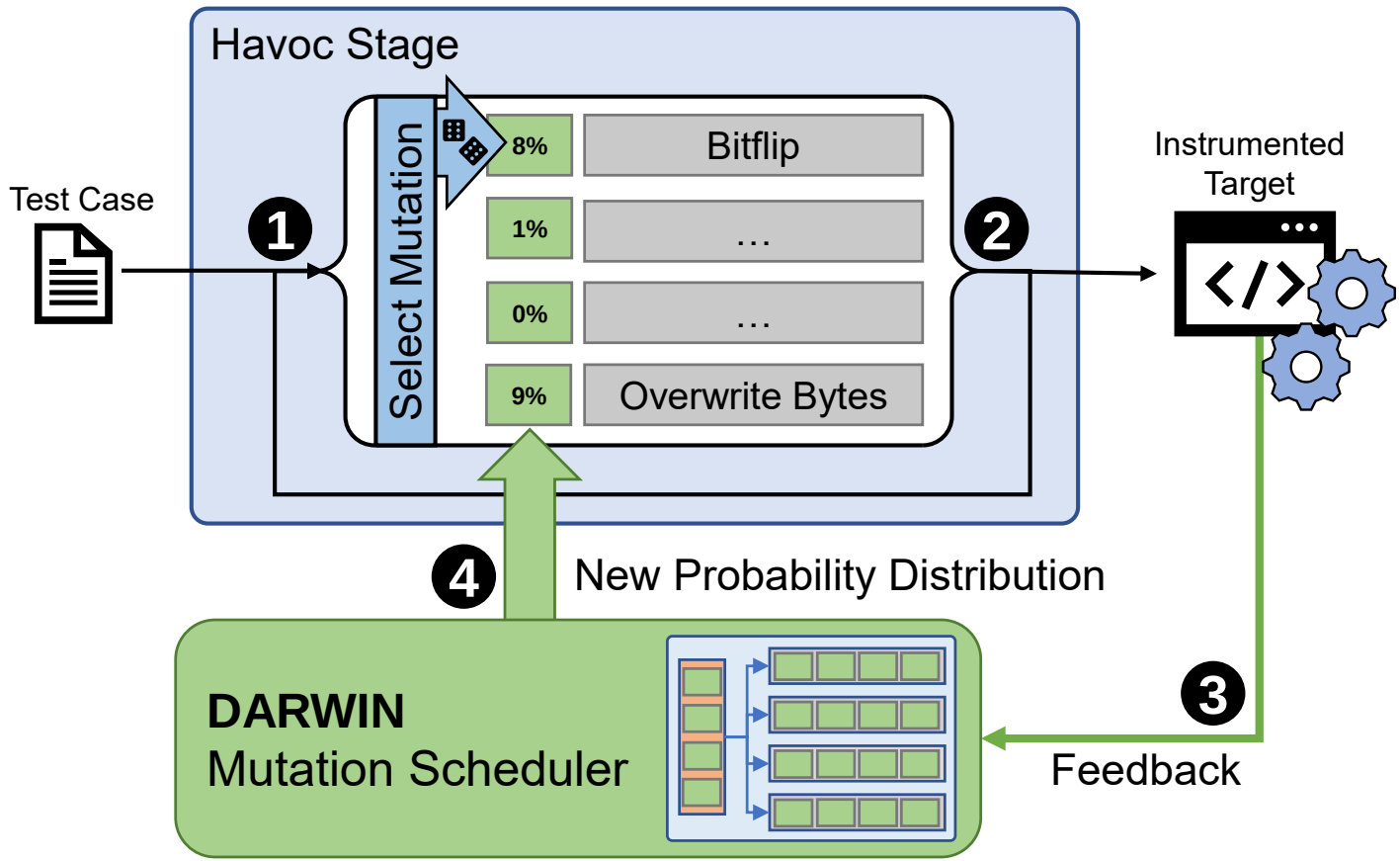
Fuzzing

- Fuzzing (fuzz testing): automated software testing technique.
- Generating inputs and feeding them to the program being tested in the hope of evoking erroneous behavior or increasing code coverage.
- *Mutation-based fuzzing*: uses a dataset of test cases (a corpus):
  - selects a test case,
  - modifies it by applying *mutation operators*,
  - feeds it to the tested program.
- Example mutation operators: bit flip, random byte value, set byte to *interesting* value, insert byte, delete byte, ...

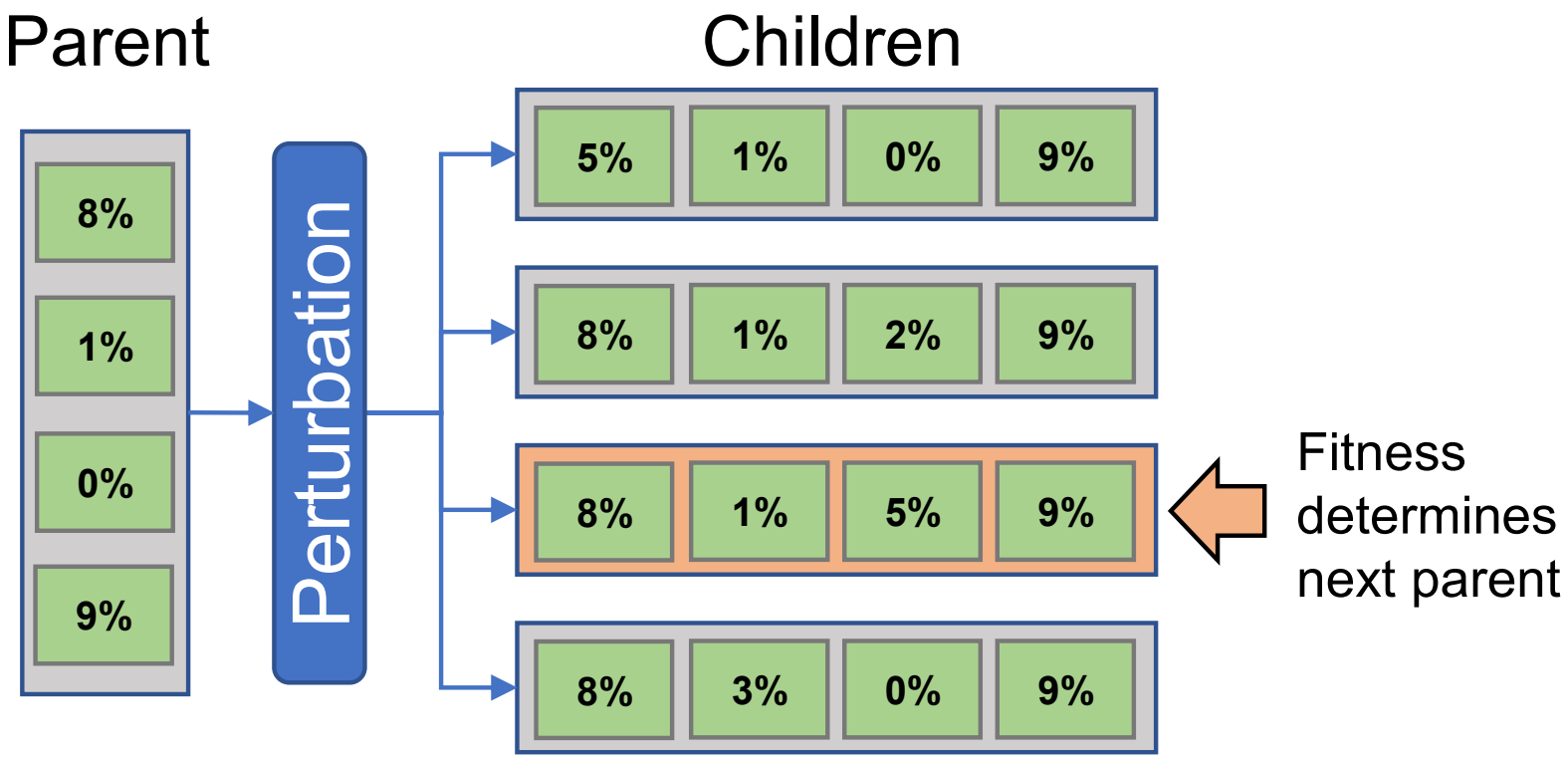
Fuzzing

- Optimization of fuzzing: finding what to fuzz or an appropriate sequence of mutation operators (*mutation scheduling*).
- Metaheuristics represents a common approach.
- Finding a practical trade-off between complexity and algorithmic improvements is challenging.
- If the evolutionary algorithm has many parameters (i.e., not easy to tune), this provides additional challenges.

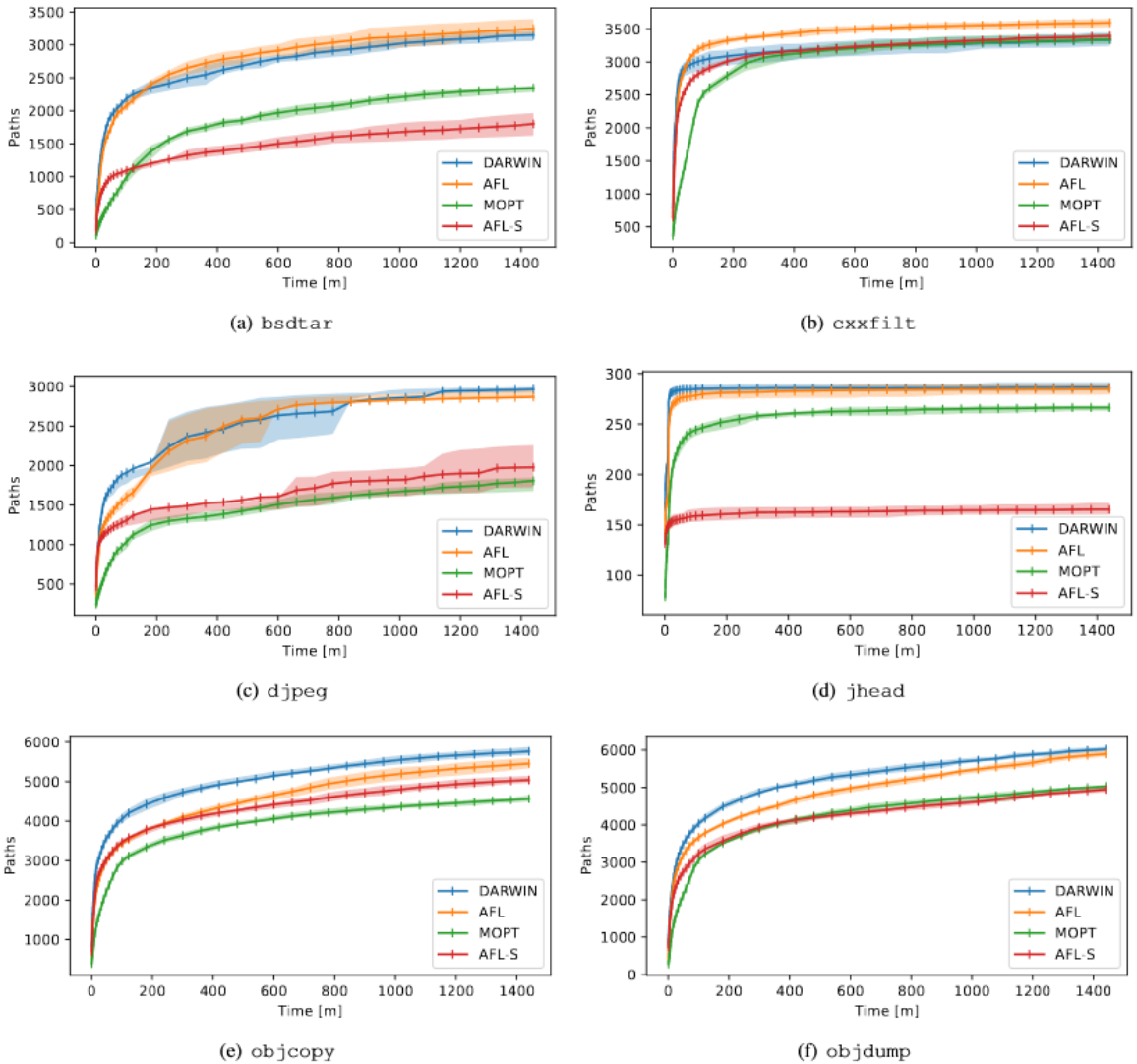
DARWIN Design



Evolution Strategy in DARWIN



Results



(a) bsdtar (b) cxxfilt

(c) djpeg (d) jhead

(e) objcopy (f) objdump

Backdoor Attacks

- Backdoor attacks are a threat where malicious samples containing a trigger are included in the dataset at training time.
- After training, a backdoor model correctly performs the main task at test time while achieving misclassification when the input contains the trigger.

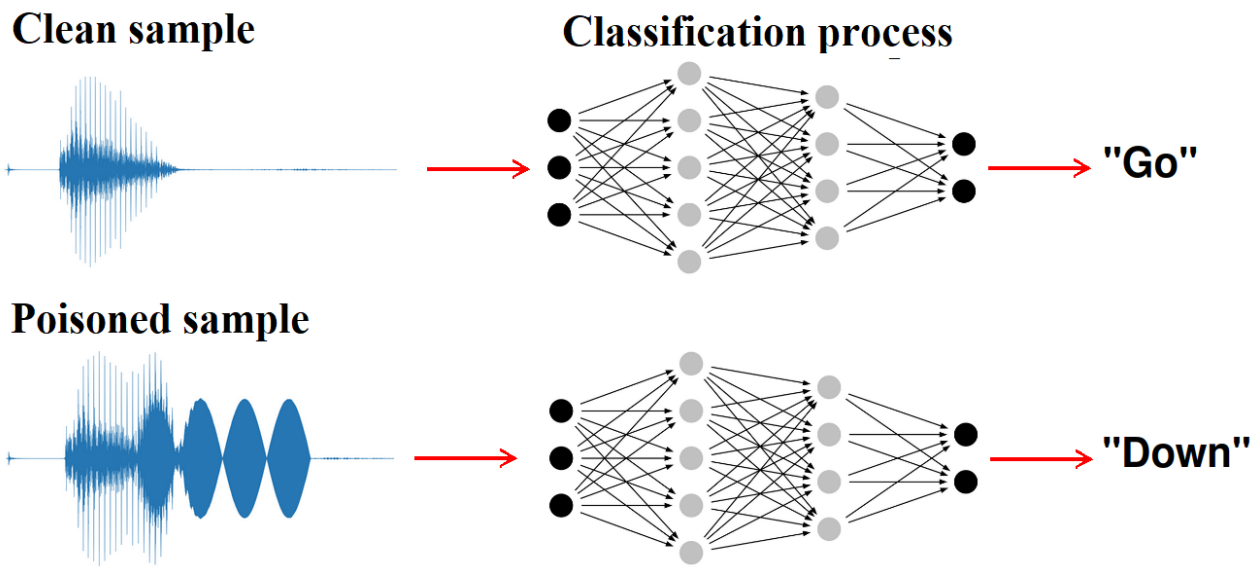
Spiking Neural Networks

- Training a well-performing DNN can be time and energy-expensive as it requires tuning many parameters with large training data.
- SNN can significantly reduce the energy consumption of DNN.
- SNN can be more robust to noise and perturbations, making them more reliable in real-world situations.
- SNN commonly operates on neuromorphic data, a time-encoded representation of the illumination changes of an object/subject captured by a DVS camera.

Inaudible Backdoor Attacks

- Automatic speech recognition (ASR) has gained much attention in recent years as it can be a very efficient form of communication between people and machines.
- Backdoor attacks are a serious threat to neural networks.
- We consider backdoor attacks on ASR systems using an inaudible trigger.
- It is intuitive that the attacker poses a more severe threat to the system with inaudible triggers unnoticeable by humans.

Inaudible Backdoor Attacks



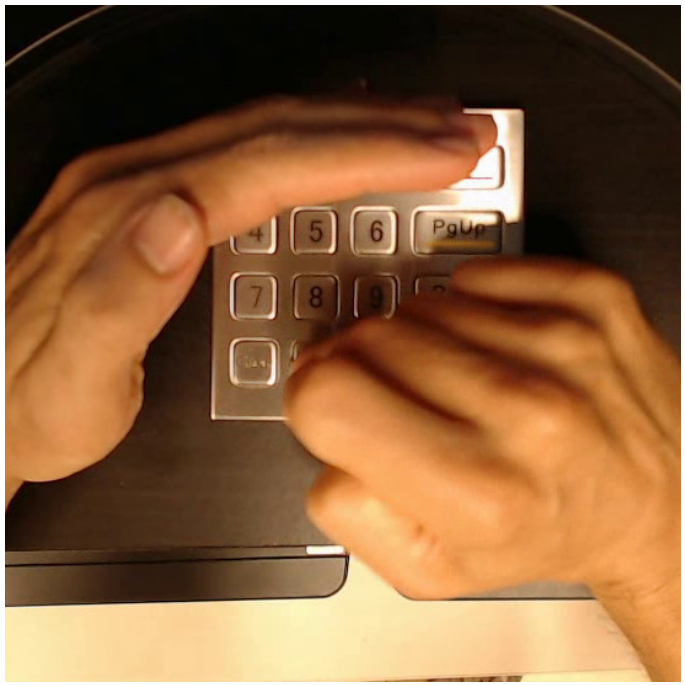
ATM Data

- We collected the environmental audio (exploiting the webcam microphone) and the keylogs of the PIN pad through the USB interface during the experiment.

Figure: Our experimental setup.

Results

- We conducted the experiments on both 4-digits and 5-digits PINs (for test set we use only well-covered PINs).



(a) *Badly covered PIN that we excluded from the validation and test tests.*

(b) *Covered PIN, where there is no direct view of the pressed key and the surrounding digits.*

Results

(a) *True digit = 7 Pred = 7 (0.999), 4 (0.000), 8 (0.000)*

(b) *True digit = 3 Pred = 3 (0.979), 2 (0.012), 6 (0.005)*

(c) *True digit = 6 Pred = 6 (0.819), 9 (0.170), 8 (0.009)*

(d) *True digit = 3 Pred = 3 (0.809), 2 (0.092), 5 (0.069)*

(e) *True digit = 3 Pred = 2 (0.329), 3 (0.315), 6 (0.185)*

Figure: PIN 73633 entered by a user in our test set in the *Single PIN pad* scenario. Our algorithm suggests 73632 as the most probable PIN (probability = 21.32%), 73633 as the second most probable PIN (probability = 20.43%), and 73636 as the third most probable PIN (probability = 11.96%). The algorithm predicts the correct PIN in the second attempt.

Outline

Expecting Too Much?

- Issues with explainability.
- Using “cool” techniques although we do not need them.

Outline

Expecting Impossible?

- Considering that AI is a silver bullet for all problems.
- Inventing and reinventing the wheel (but with a new name).
- Hyperproduction of techniques and papers.

Outline

Conclusions

- AI is a goldmine for security and privacy research.
- There are many successful examples where AI achieved results not possible with other techniques.
- Unfortunately, there are also many bad examples.
- AI is a tool that can help a lot, but it also opens some new challenges.
- Like for any couple, there are some rough periods.
- But will stay a couple for many years to come.