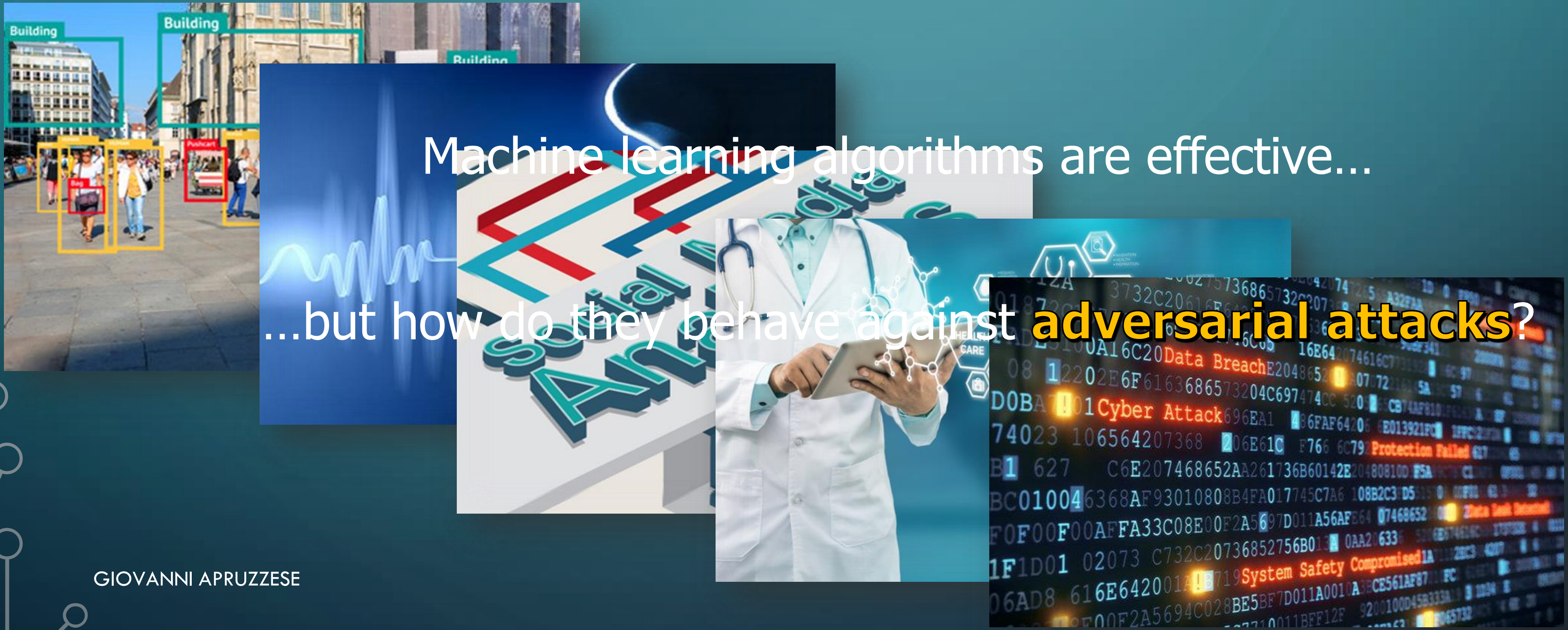
Evading Botnet Detectors based on Flows and Random Forest with Adversarial Samples

Giovanni Apruzzese, Michele Colajanni

University of Modena and Reggio Emilia, Italy

CONTEXT: MACHINE LEARNING

The popularity of machine learning is skyrocketing.



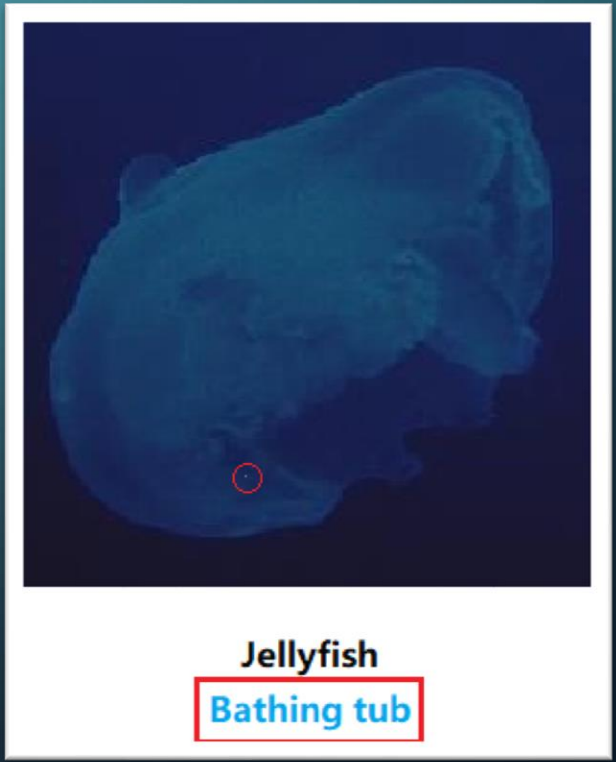
Machine learning algorithms are effective…

…but how do they behave against **adversarial attacks**?

CONTEXT: ADVERSARIAL ATTACKS

Adversarial attacks involve the creation of specific samples with the goal of thwarting the machine learning algorithm.

Even **tiny perturbations** can **greatly affect** the prediction performance



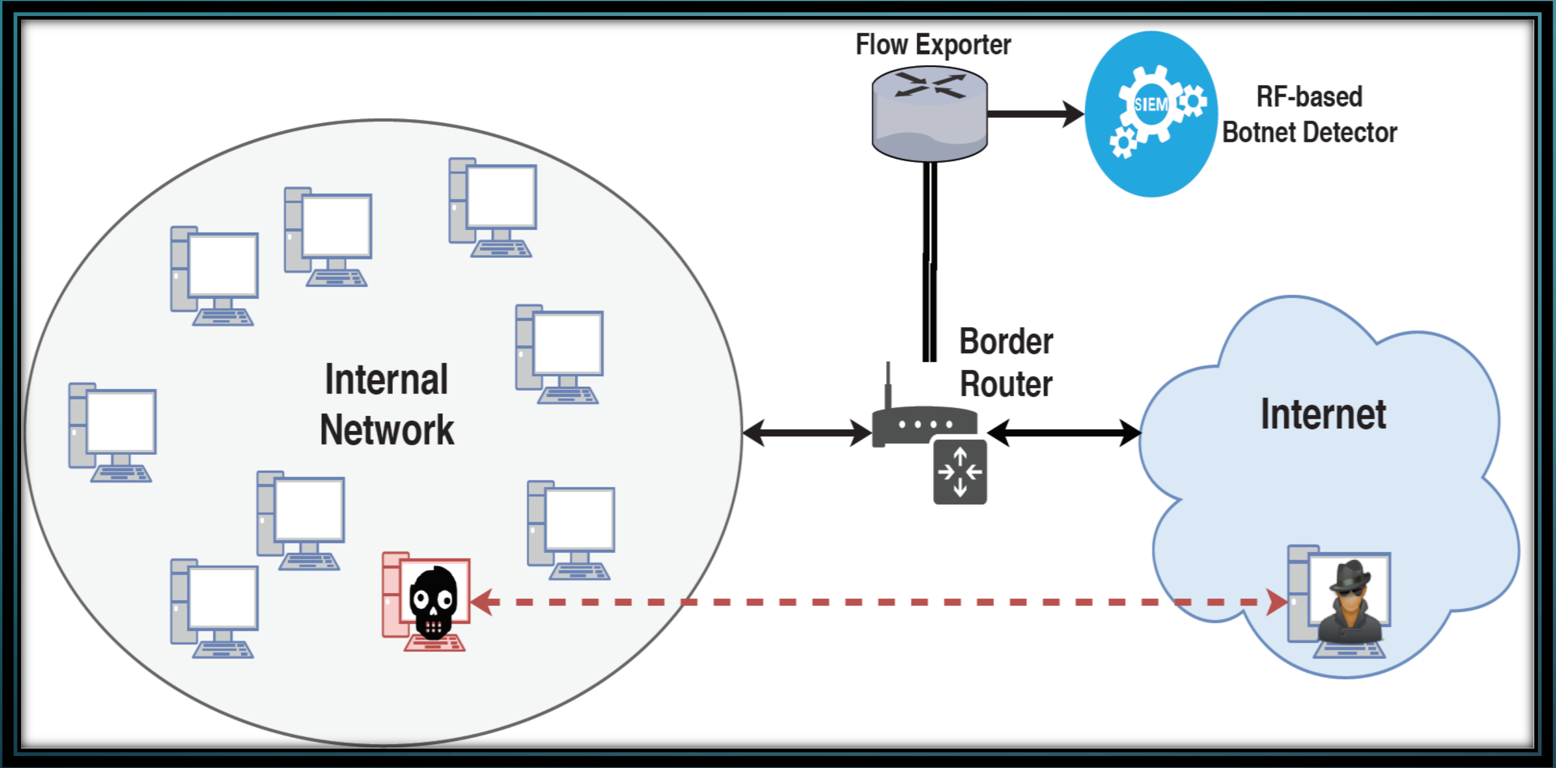
- Rich research area within the image processing field…
- …but comprehensive analyses from a **cybersecurity** perspective are scarce.

CONTRIBUTION & MOTIVATION

We present an empirical evaluation of adversarial attacks against a **flow-based botnet detector** that leverages the **random forest** algorithm.

Flow-based	• Growing practice for network intrusion detection • Several advantages w.r.t. traditional PCAP
Botnet detector	• Botnets still represent a dangerous threat
Random Forest	• Considered as one of the best algorithms for network intrusion detection tasks

APPLICATION SCENARIO



Attacker Model

- Goal: evade the botnet detector
- Knowledge: Limited
- Capabilities: Limited
- Strategy: alter the bot(s) communications

Realistic assumptions

EXPERIMENTS – OUTLINE

1. Develop a botnet detector with good performance
2. Generate **realistic** adversarial samples
3. Evaluate the detector against the generated adversarial samples

EXPERIMENTS – DATASET

CTU Dataset

- Public dataset of labelled network flows containing botnet traffic
- Dozens of internal hosts
- Over 20M of netflows, corresponding to more than 850M packets
- Contains botnet traffic generated by multiple malware families:

  Neris, Rbot, Virut, Menti, Murlo, NSIS.ay

Realistic use-case

EXPERIMENTS – BASELINE RESULTS

- We first train and test the botnet detector on the **unmodified samples**:

Malware	FP rate	FN rate	Precision	DR
Neris	0.0014	0.0472	0.9624	0.9528
Rbot	< 0.0001	0.0015	0.9999	0.9985
Virut	0.0003	0.0525	0.9871	0.9475
Menti	0	0.0015	1	0.9967
Murlo	0	0.0162	1	0.9838
NSIS.ay	< 0.0001	0.1557	0.9872	0.8443

- These results show that the detector obtains **appreciable performance**…

EXPERIMENTS – GENERATING ADVERSARIAL SAMPLES

Goal: generate adversarial samples by introducing small modifications into the malicious flow samples

Procedure:

1. Create one *malicious* dataset for each malware family
2. For each malicious dataset, generate multiple adversarial datasets:
   a) Select several *groups of features*
   b) For each group, increase the values of its features through multiple *steps*

EXPERIMENTS – GENERATING ADVERSARIAL SAMPLES

Group	Altered features
1a	Duration (s)
1b	Src_bytes
1c	Dst_bytes
1d	Tot_pkts
2a	Duration, Src_bytes
2b	Duration, Dst_bytes
2c	Duration, Tot_pkts
2e	Src_bytes, Tot_pkts
2d	Src_bytes, Dst_bytes
2f	Dst_bytes, Tot_pkts
3a	Duration, Src_bytes, Dst_bytes
3b	Duration, Src_bytes, Tot_pkts
3c	Duration, Dst_bytes, Tot_pkts
3d	Src_bytes, Dst_bytes, Tot_pkts
4a	Duration, Src_bytes, Dst_bytes, Tot_pkts

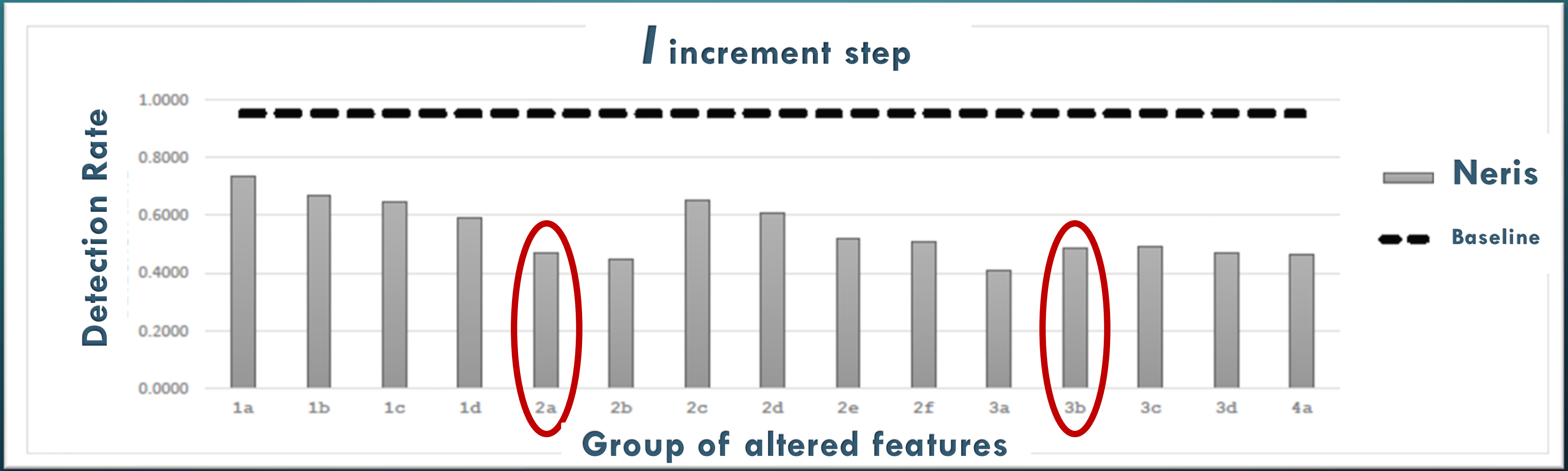
tes	Tot_pkts
	+1
	+2
	+5
	+10
	+15
	+20
	+30
	+50
4	+100

EXAMPLE

- With the *I* … *durations* a… all its
- The advers… outgoing b… its durations,

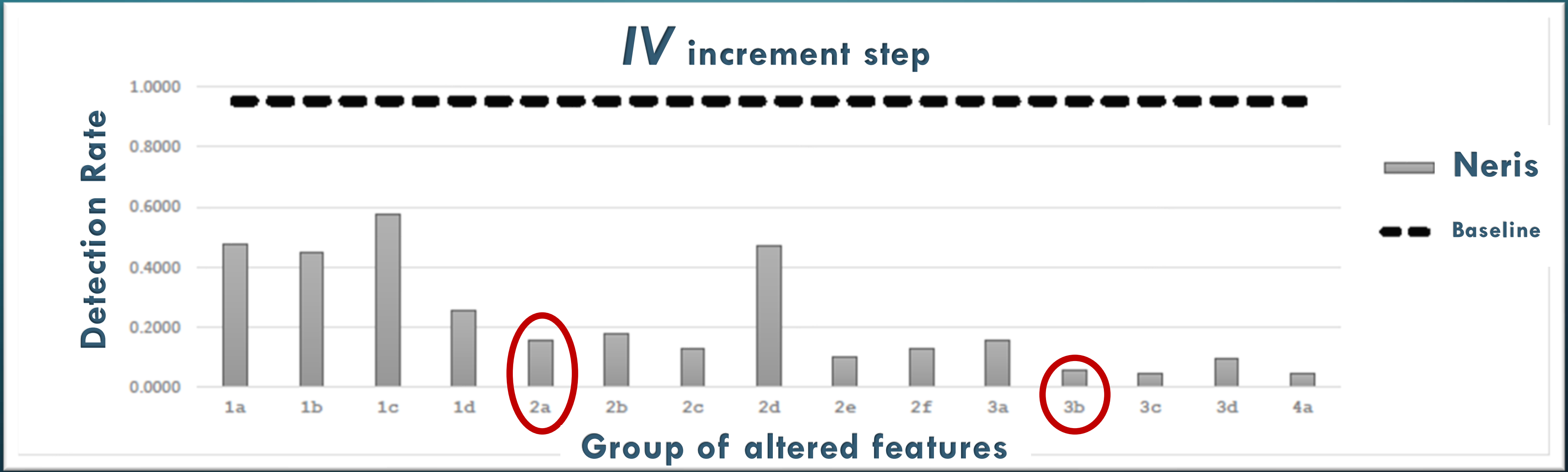
EXPERIMENTS – ADVERSARIAL ATTACKS RESULTS

- ...but the situation changes when tested against the **adversarial samples**:



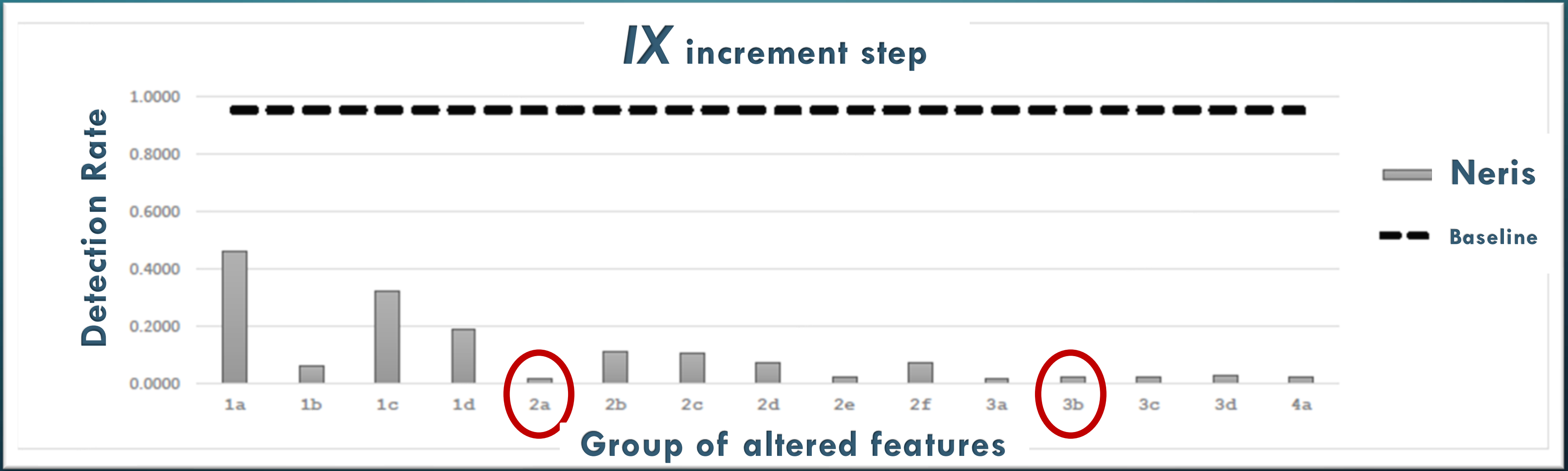
EXPERIMENTS – ADVERSARIAL ATTACKS RESULTS

- ...and it only gets worse...



EXPERIMENTS – ADVERSARIAL ATTACKS RESULTS

- ...and worse:



CONCLUSION

- The adoption of machine learning algorithms is constantly growing.
- These techniques need to be evaluated against **adversarial attacks**, especially from a cybersecurity perspective.
- We expose the fragility against adversarial perturbations of *flow-based botnet detectors* relying on the *random forest* algorithm.

Extensive experimental evaluation shows that the **detection rate** of a similar detector drops to values **lower than 1%** just by introducing small and targeted modifications to the network communications of the infected machine.

CONCLUSION – POSSIBLE SOLUTIONS

- Re-training with adversarial samples (**Adversarial Learning**)

Requires the **availability** and **mainteance** of a **realistic** adversarial dataset

- Use different features that cannot be modified by the attacker

Decreases the performance of the detector against unmodified samples

Evading botnet detectors based on flows and Random Forest with adversarial samples

Giovanni Apruzzese

PhD. Student

University of Modena, Italy

giovanni.apruzzese@unimore.it

FOLLOW UP:

HARDENING RANDOM FOREST DETECTORS THROUGH DISTILLATION

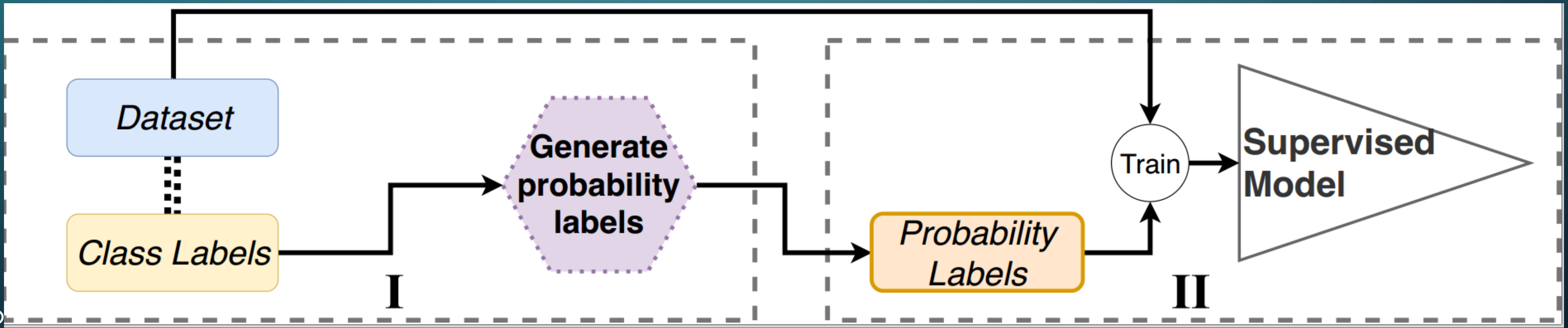
- Cyber Detectors employing rigid classification criteria may be more vulnerable to subtle adversarial perturbations.
- Existing detectors are trained through *class labels* that separate samples in disjointed categories.
- **The cyber domain is intrinsically fuzzy, and a sample may present characteristics belonging to different categories.**

We aim to introduce some degree of flexibility and uncertainty by using *probability labels*

PROBLEM ANALYSIS

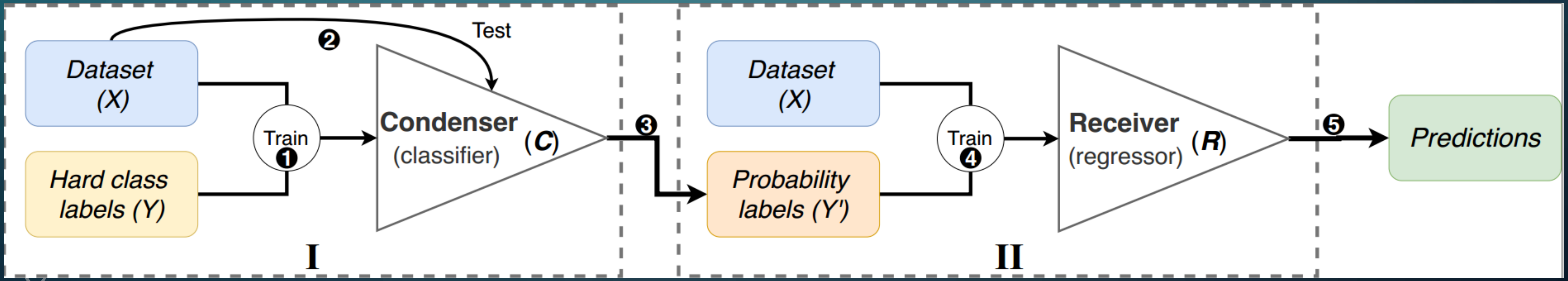
- In the cyber domain, probability labels are not readily available.

- → We devise an original solution that is built upon two phases:
  I. Generation of probability labels from hard class labels;
  II. Deployment of a supervised model trained with the generated probability labels to perform the cyber detection.



APPLICATION TO THE RANDOM FOREST ALGORITHM

- The initial phase is performed through a random forest classifier **(Condenser)**.
  - We first train this classifier with the hard-class labels.
  - We leverage the intrinsic property of the random forest algorithm of being an ensemble method: we generate the probability vectors by considering the percentage of estimators that predicted a particular result.
- In the second phase, the probability vectors are used as training labels for a random forest regressor **(Receiver).**

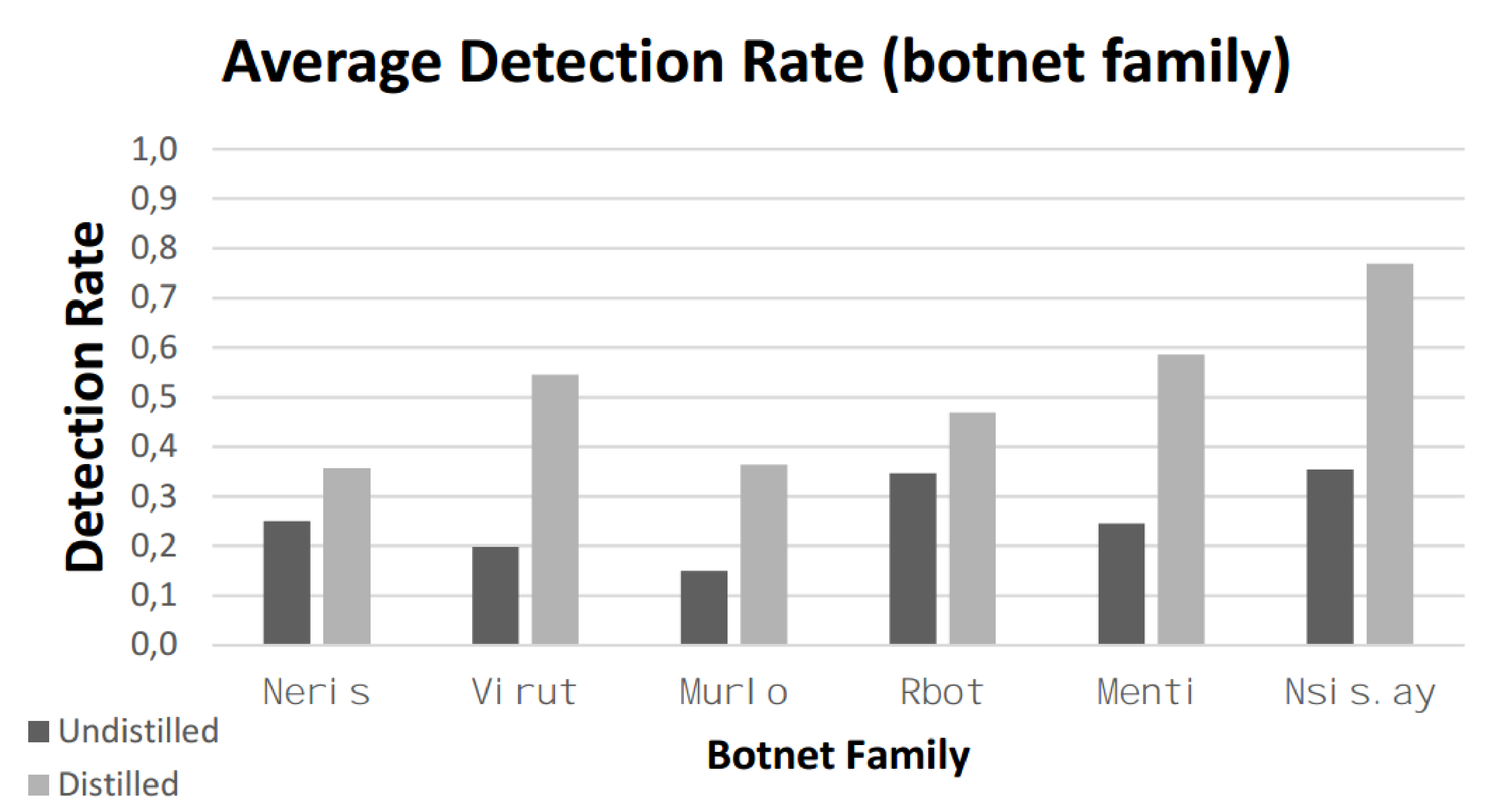


RESULTS IN NON-ADVERSARIAL SETTINGS

Table VI: Baseline vs. Distilled model performance.

Botnet	Instance type	F1-Score	Precision	Recall	FPR	TNR	FNR
Neris	Undistilled	0.9577	0.9615	0.9540	0.0015	0.9985	0.0461
	Distilled	0.9651	0.9671	0.9632	0.0013	0.9987	0.0368
Virut	Undistilled	0.9682	0.9876	0.9496	0.0002	0.9998	0.0504
	Distilled	0.9753	0.9876	0.9633	0.0002	0.9998	0.0367
Murlo	Undistilled	0.9932	1	0.9866	0	1	0.0134
	Distilled	0.9968	1	0.9937	0	1	0.0063
Rbot	Undistilled	0.9994	0.9999	0.9999	< 0.0001	1	0.0010
	Distilled	0.9995	0.9999	0.9990	< 0.0001	1	0.0010
Menti	Undistilled	0.9984	1	0.9969	0	1	0.0031
	Distilled	0.9979	0.9997	0.9969	< 0.0001	1	0.0031
NSIS.ay	Undistilled	0.9213	0.9925	0.8596	< 0.0001	1	0.1404
	Distilled	0.9273	0.9784	0.8812	0.0001	0.9999	0.1188
Average	Undistilled	0.9729	0.9774	0.9684	0.0005	0.9995	0.0315
	Distilled	0.9777	0.9804	0.9751	0.0004	0.9996	0.0249

RESULTS IN ADVERSARIAL SETTINGS



CONCLUSION

- Detection models based on machine learning have features that are too sensitive to adversarial perturbations.
- The proposed solution allows to develop detectors that:
  - **achieve <u>same or better detection performance</u> than existing algorithms in non-adversarial scenarios;**
  - **with <u>improved robustness</u> against adversarial attacks.**
- There is still space for researches that aim to further improve the detection rates.