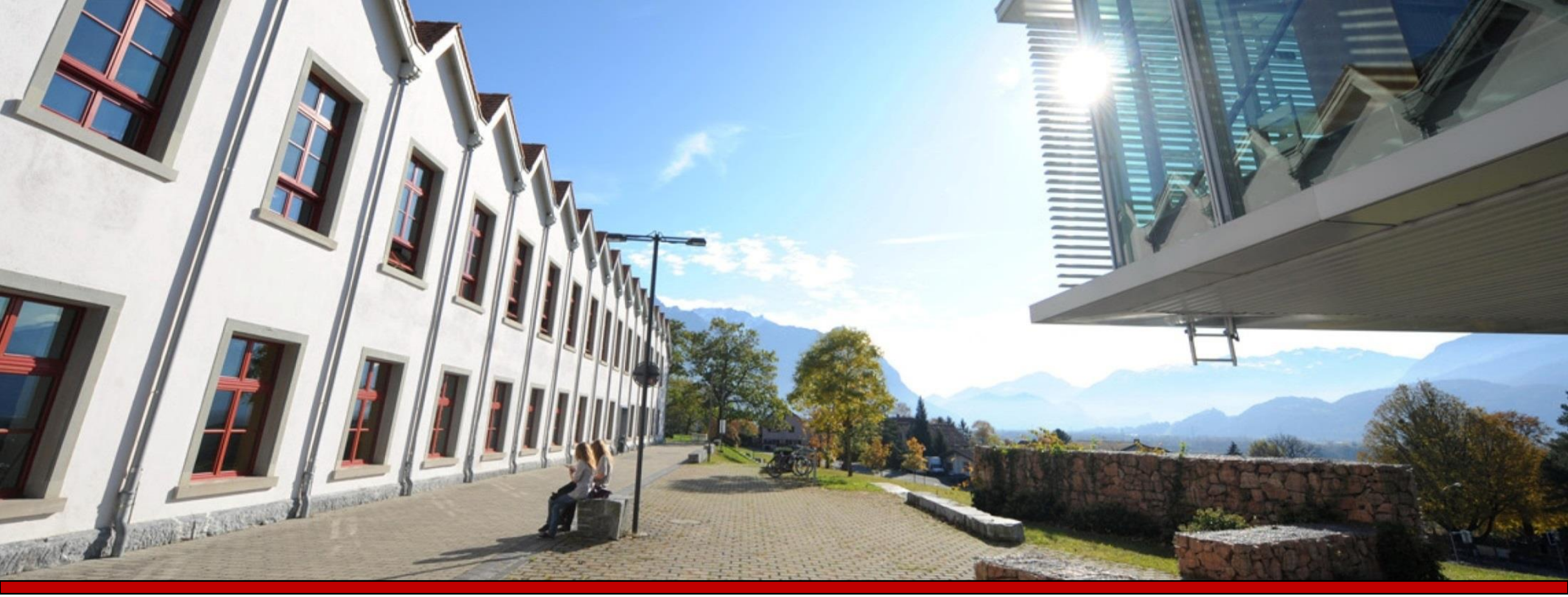


The relationship between **Machine Learning & Cybersecurity**

Giovanni Apruzzese, PhD

TU Delft – May 3rd, 2022

UNIVERSITÄT LIECHTENSTEIN

whoami: Dr. **Giovanni Apruzzese**

- **Background:**
  - Did my academic studies (BSc, MSc, PhD) at University of Modena, Italy.
    - Supervisor: Prof. Michele Colajanni
  - In 2019, spent 6 months at Dartmouth College, USA.
    - Supervisor: Prov. VS Subrahmanian
  - Joined the University of Liechtenstein in July 2020 as a PostDoc Researcher.
    - Supervisor: Prof. Pavel Laskov
  - Met Prof. Mauro Conti in 2019, with whom I have been collaborating since 2020.

- **Interests:**
  - Cybersecurity, machine learning, and any network-related topic (+🎮)
  - I like talking, researching and teaching – in a “pragmatic” way ☺

- **Contact information**:
  - Work Email: giovanni.apruzzese@uni.li
  - Feel free to contact me if you have any questions.
    - I reply fast, and will happily do so!

What I do

Machine Learning + Cybersecurity

- Applying ML to *provide security* of a given information system
  - E.g.: using ML to detect network intrusions
- *Attacking / Defending* ML applications
  - E.g.: evading a ML model that detects phishing websites
- Using machine learning *offensively* against another system
  - E.g.: artificially generating “fake” images

BONUS

- Using ML to attack a security system based on ML

Outline of Today

- Fundamentals of Machine Learning and Cybersecurity

- Using unlabelled data for Machine Learning in Cyberthreat Detection
  - Ref: Giovanni Apruzzese, Luca Pajola, and Mauro Conti. "The Cross-evaluation of Machine Learning-based Network Intrusion Detection Systems." IEEE Transactions on Network and Service Management (2022).

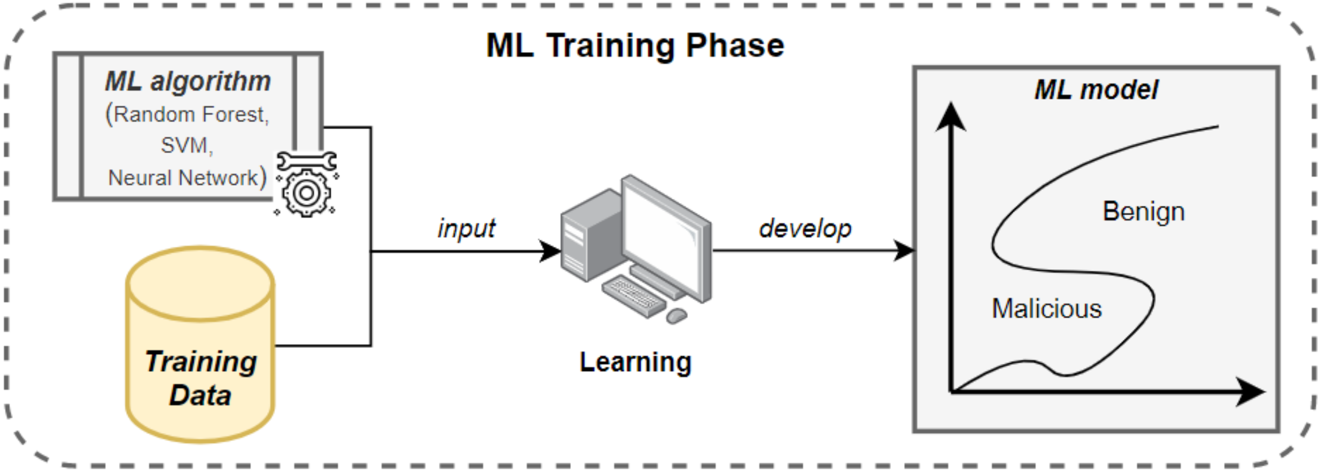
- Improving Machine Learning in Network Intrusion Detection
  - Ref: Giovanni Apruzzese, Pavel Laskov, Aliya Tastemirova. “SoK: The Impact of Unlabelled Data for Cyberthreat Detection.” IEEE European Symposium on Security and Privacy (2022).

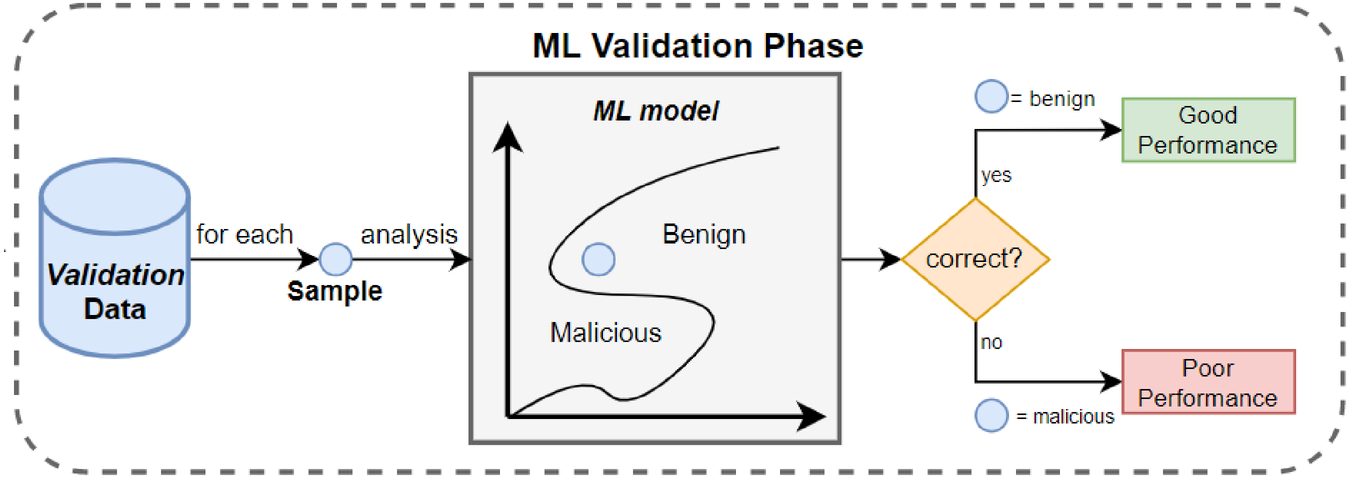
- The security of Machine Learning-based Phishing Website Detectors
  - Ref: Giovanni Apruzzese, Mauro Conti, Ying Yuan. “SpacePhish: The Evasion-space of Adversarial Attacks against Phishing Website Detectors”. TBD

- Adversarial Attacks against Humans **and** Machine Learning
  - Ref: Johannes Schneider, Giovanni Apruzzese. “Concept-based Adversarial Attacks: Tricking Humans and Classifiers alike.” IEEE Symposium on Security and Privacy – Deep Learning and Security Workshop (2022)

Fundamentals of Machine Learning and Cybersecurity

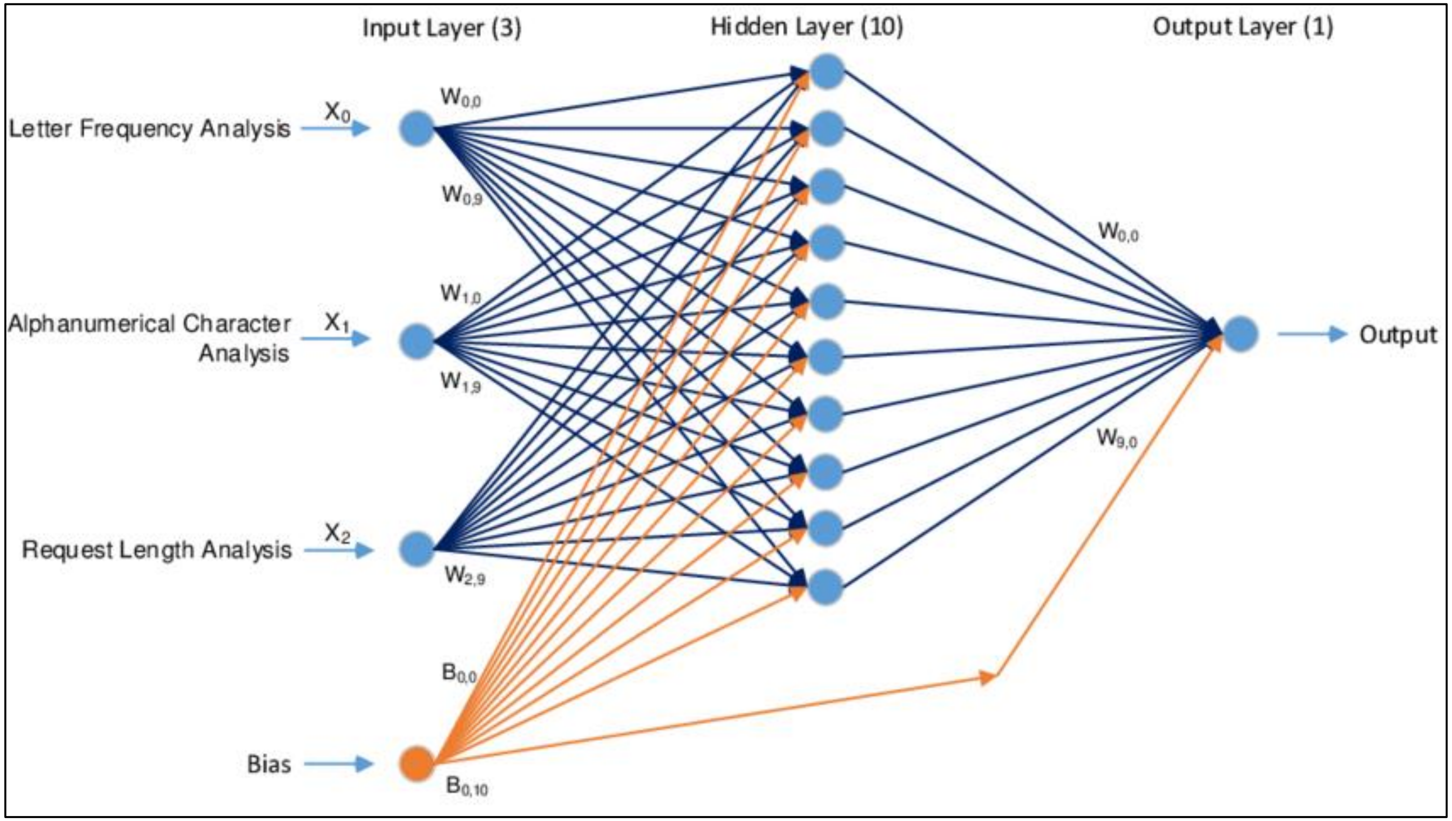
Machine Learning workflow: Training and Testing

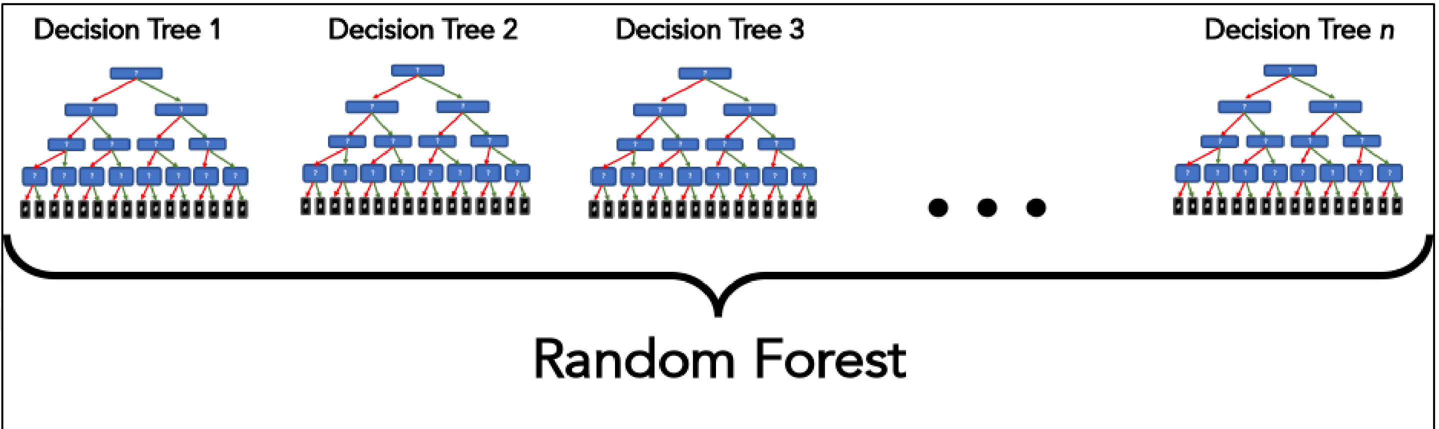




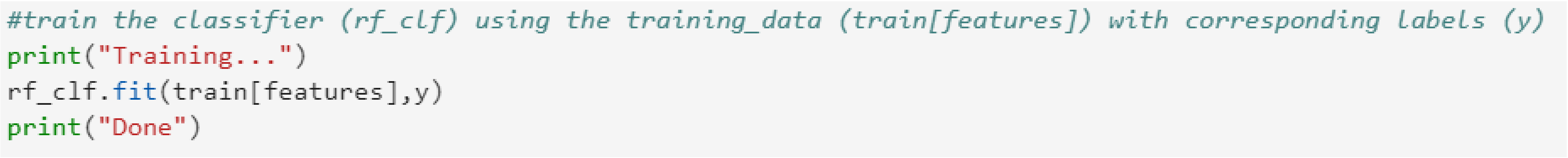
UNIVERSITÄT
LIECHTENSTEIN

Question: do you think that training ML models is difficult?





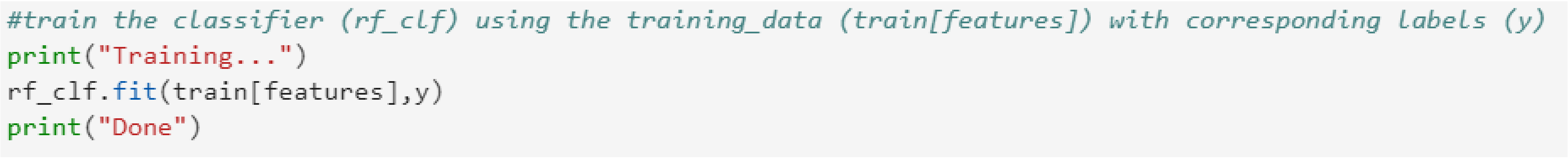
Question: do you think that training ML models is difficult?



```
#train the classifier (rf_clf) using the training_data (train[features]) with corresponding labels (y)
print("Training...")
rf_clf.fit(train[features],y)
print("Done")
```

Question: do you think that training ML models is difficult?

PROBLEMS (data)



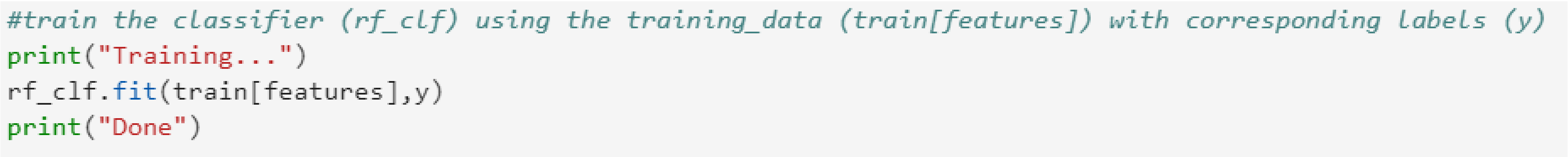
```
#train the classifier (rf_clf) using the training_data (train[features]) with corresponding labels (y)
print("Training...")
rf_clf.fit(train[features],y)
print("Done")
```

PROBLEMS (tuning)

UNIVERSITÄT
LIECHTENSTEIN

Question: do you think that training ML models is difficult?

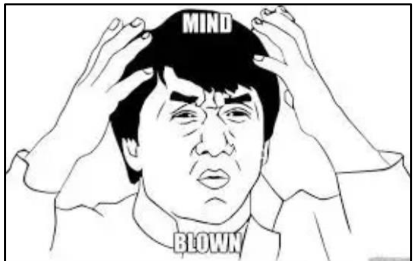
PROBLEMS (data)



```
#train the classifier (rf_clf) using the training_data (train[features]) with corresponding labels (y)
print("Training...")
rf_clf.fit(train[features],y)
print("Done")
```

PROBLEMS (tuning)

Of course, you're always free to go, learn and improve the *fit* function:
https://github.com/scikit-learn/scikit-learn/blob/baf828ca1/sklearn/ensemble/_forest.py#L297



Common issues of ML in Cybersecurity

- Applying Machine Learning requires *data* to train a ML model
- Depending on the "problem" solved by such model, the data may require *labels*
- **Obtaining (any) data has a cost, and labelled data is (very) *expensive***

- Machine Learning models are ultimately just a component within a system
- **Such ML models *can* be targeted by "Adversarial Attacks"**
- Such strategies ultimately aim to compromise the functionality of the ML model.

- The cybersecurity domain implicitly assumes the presence of attackers.
- Attackers are *human beings,* and hence operate with a *cost/benefit* mindset
- **Such considerations must be made when analyzing the security of (any) IT system**

Unlabelled data for Machine Learning in Cyberthreat Detection

Semisupervised Learning

○ Labelled data is expensive, but *unlabelled* data is cheap(er).

→ Why not using unlabelled data to improve the proficiency of ML models?

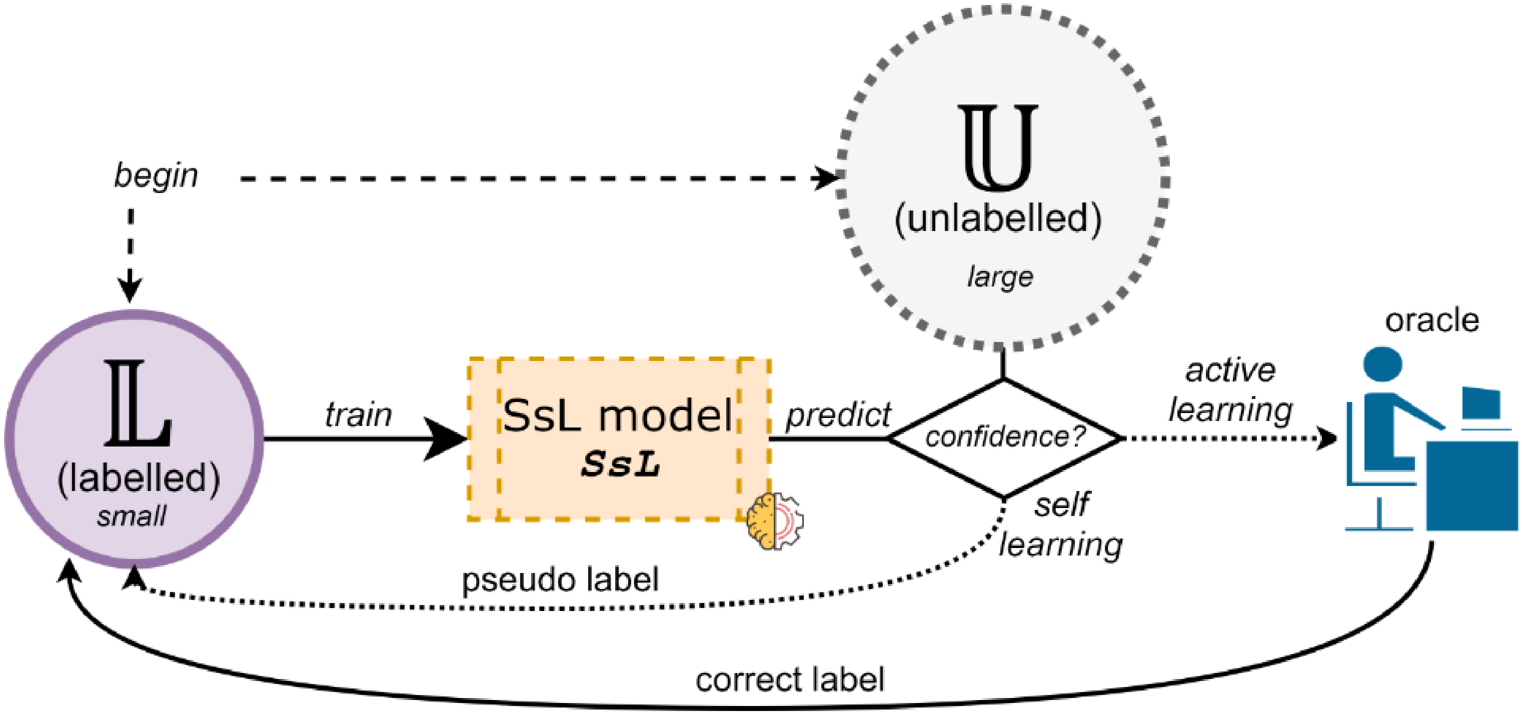
Mixing *labelled* with *unlabelled* data is a ML approach denoted as **"Semisupervised Learning" (SsL)**

Semisupervised Learning

- Labelled data is expensive, but *unlabelled* data is cheap(er).

→ Why not using unlabelled data to improve the proficiency of ML models?

Mixing *labelled* with *unlabelled* data is a ML approach denoted as **“Semisupervised Learning” (SsL)**

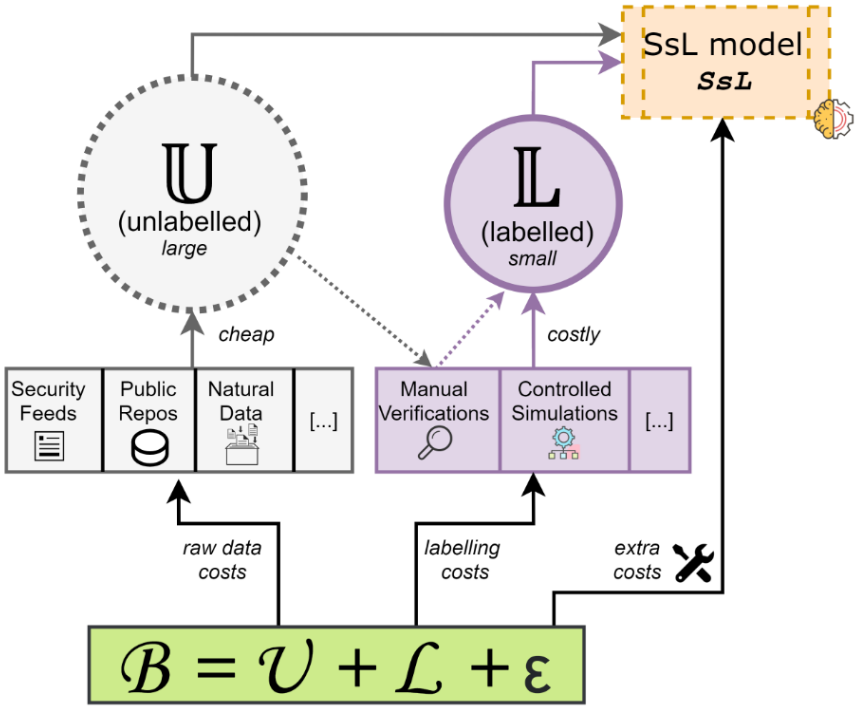


Examples of SsL: *active learning* and *self learning* (e.g., *pseudo labelling*)

Giovanni Apruzzese, *PhD*
giovanni.apruzzese@uni.li

Goal of Semisupervised Learning

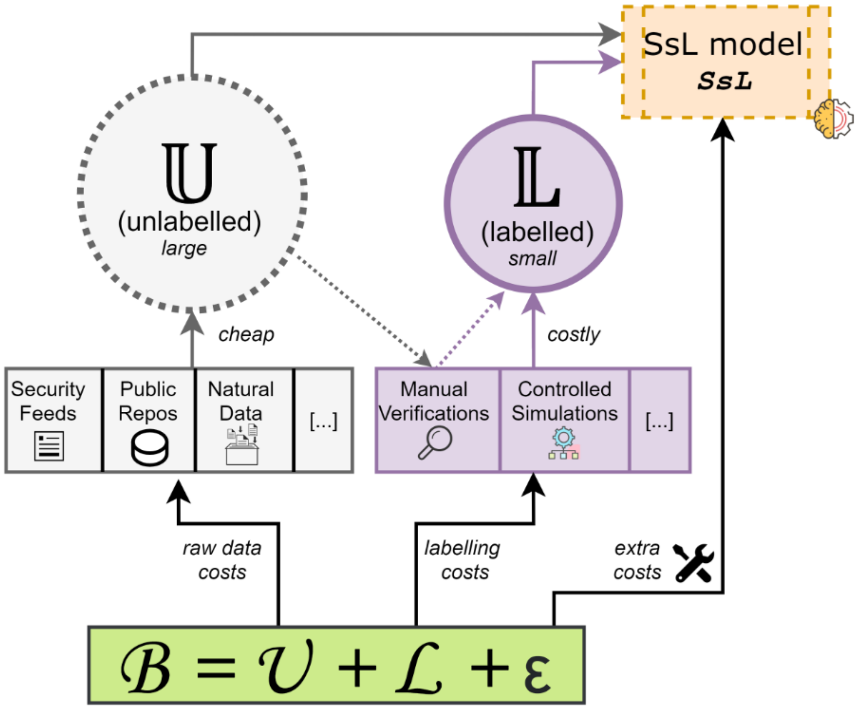
- Developing SsL models is cheaper than "supervised learning" (SL) models, **but it is not free.**

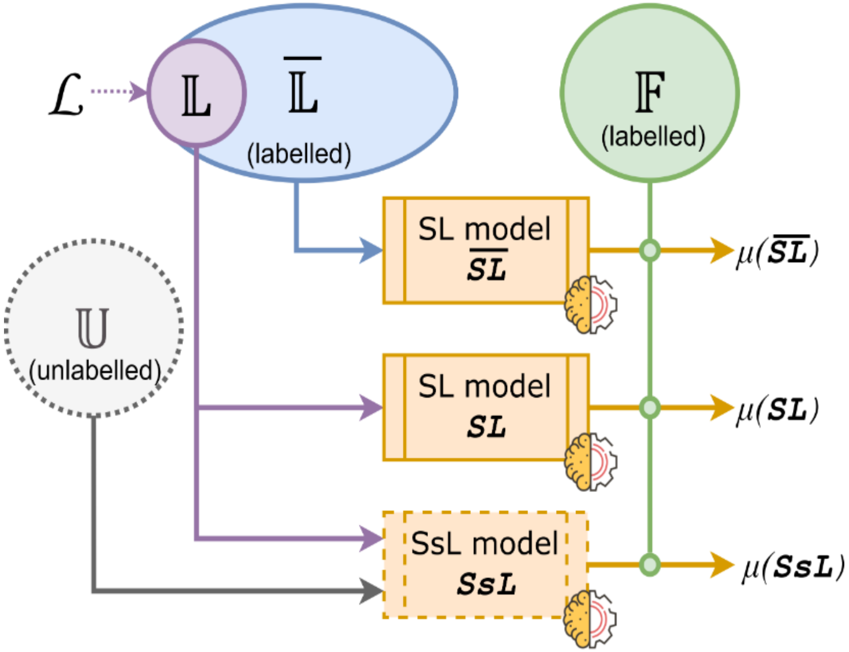


Goal of Semisupervised Learning

- Developing SsL models is cheaper than "supervised learning" (SL) models, **but it is not free.**

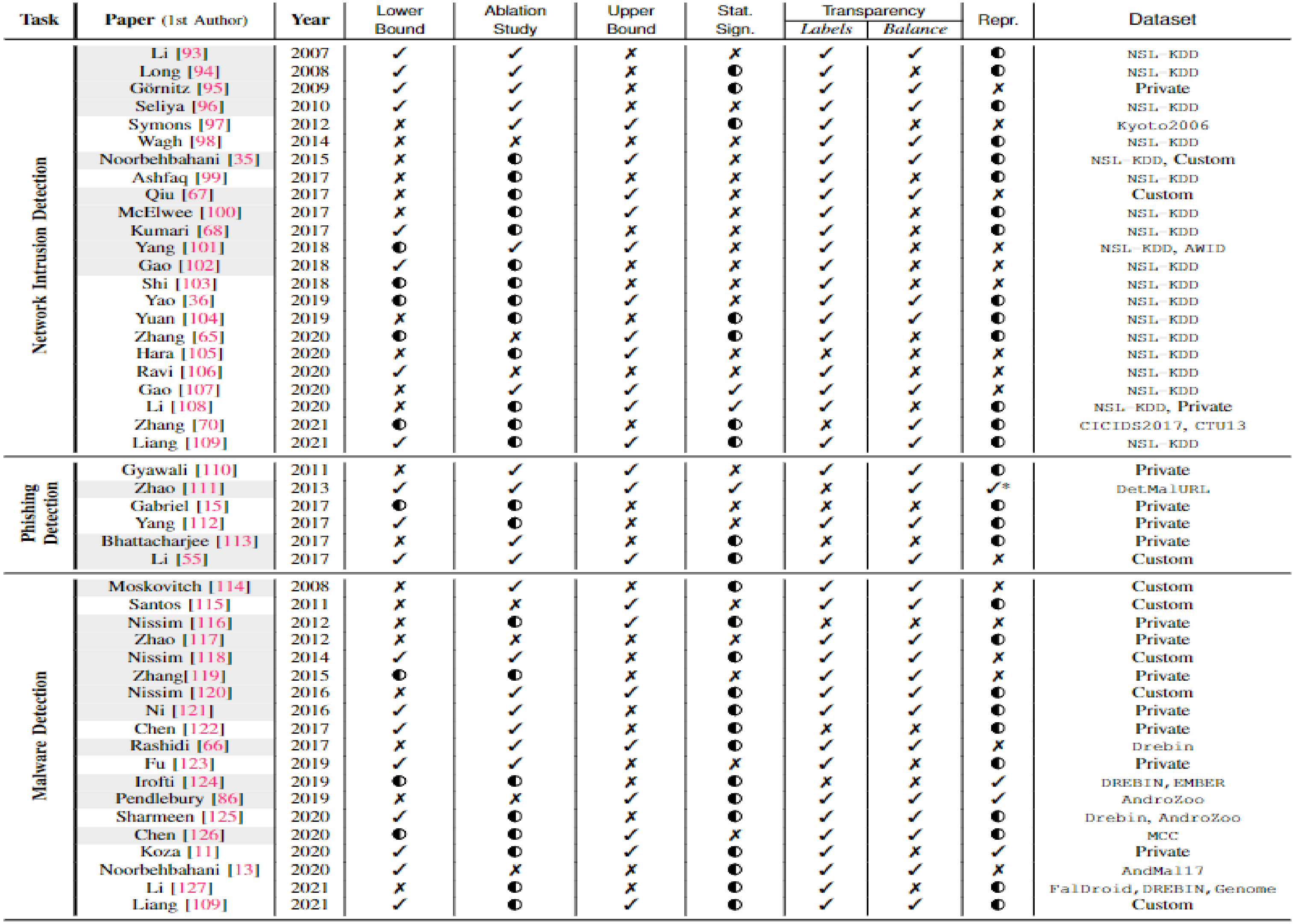
- A SsL model should achieve a *performance superior* than a SL model that uses the *same labelling budget*





Problem: nobody cares

The current state-of-the-art does not allow to determine whether SsL methods applied in Cyberthreat Detection are truly beneficial

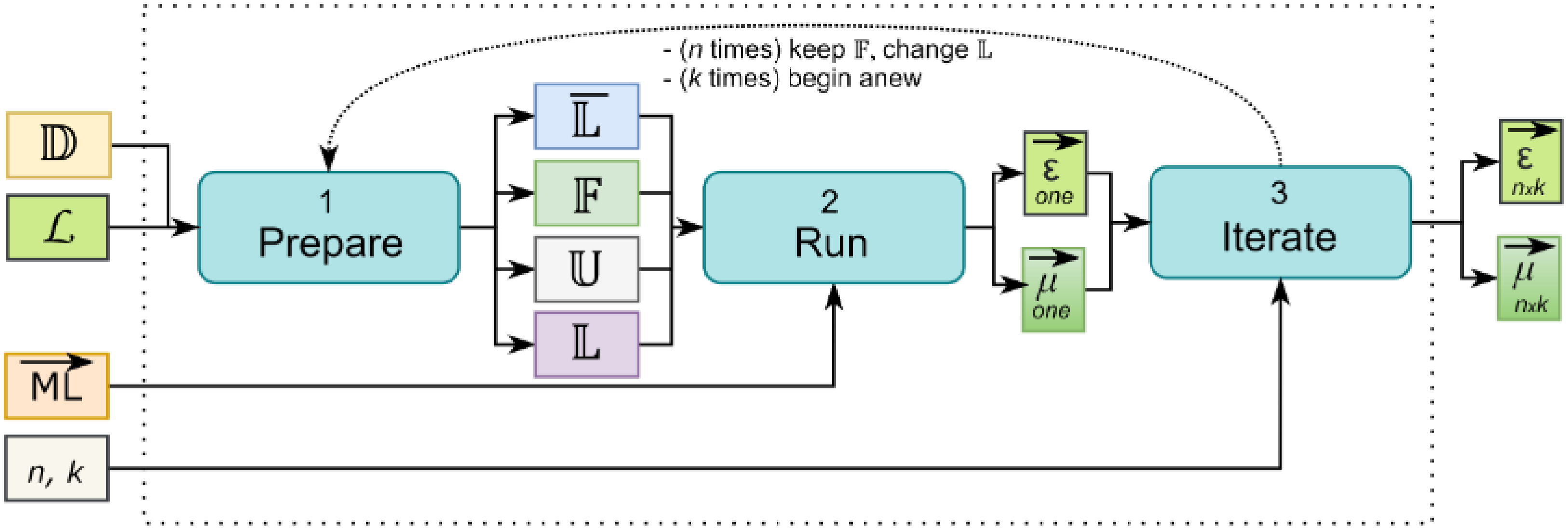


Task	Paper (1st Author)	Year	Lower Bound	Ablation Study	Upper Bound	Stat. Sign.	Transparency		Repr.	Dataset
							Labels	*Balance*		
Network Intrusion Detection	Li [93]	2007	✓	✓	✗	✗	✓	✓	◐	NSL-KDD
	Long [94]	2008	✓	✓	✗	◐	✓	✗	◐	NSL-KDD
	Görnitz [95]	2009	✓	✓	✗	◐	✓	✓	✗	Private
	Seliya [96]	2010	✓	✓	✗	✗	✓	✓	◐	NSL-KDD
	Symons [97]	2012	✗	✓	✓	◐	✓	✗	✗	Kyoto2006
	Wagh [98]	2014	✗	✗	✗	✗	✓	✓	◐	NSL-KDD
	Noorbehbahani [35]	2015	✗	◐	✓	✗	✓	✓	◐	NSL-KDD, Custom
	Ashfaq [99]	2017	✗	◐	✗	✗	✓	✗	◐	NSL-KDD
	Qiu [67]	2017	✗	◐	✓	✗	✓	✓	✗	Custom
	McElwee [100]	2017	✗	◐	✓	✗	✓	✗	◐	NSL-KDD
	Kumari [68]	2017	✓	◐	✗	✗	✓	✗	◐	NSL-KDD
	Yang [101]	2018	◐	✓	✓	✗	✓	✗	✗	NSL-KDD, AWID
	Gao [102]	2018	✓	◐	✗	✗	✓	✗	✗	NSL-KDD
	Shi [103]	2018	◐	◐	✗	✗	✓	✗	✗	NSL-KDD
	Yao [36]	2019	◐	◐	✓	✗	✓	✓	◐	NSL-KDD
	Yuan [104]	2019	✗	◐	✗	◐	✓	✓	◐	NSL-KDD
	Zhang [65]	2020	◐	✗	✓	◐	✓	✗	◐	NSL-KDD
	Hara [105]	2020	✗	◐	✓	✗	✗	✗	✗	NSL-KDD
	Ravi [106]	2020	✓	✗	✗	✗	✓	✗	✗	NSL-KDD
	Gao [107]	2020	✗	✓	✓	✓	✓	✓	✗	NSL-KDD
	Li [108]	2020	✗	◐	✓	✓	✓	✗	◐	NSL-KDD, Private
	Zhang [70]	2021	◐	◐	✗	◐	✗	✓	◐	CICIDS2017, CTU13
	Liang [109]	2021	✓	◐	✓	◐	✓	✓	◐	NSL-KDD
Phishing Detection	Gyawali [110]	2011	✗	✓	✓	✗	✓	✓	◐	Private
	Zhao [111]	2013	✓	✓	✓	✓	✗	✓	✓*	DetMalURL
	Gabriel [15]	2017	◐	◐	✗	✗	✗	✗	◐	Private
	Yang [112]	2017	✓	◐	✗	✗	✓	✓	◐	Private
	Bhattacharjee [113]	2017	✗	✓	✗	◐	✗	✗	◐	Private
	Li [55]	2017	✓	✓	✓	◐	✓	✓	✗	Custom
Malware Detection	Moskovitch [114]	2008	✗	✓	✗	◐	✓	✓	✗	Custom
	Santos [115]	2011	✗	✗	✓	✗	✓	✓	◐	Custom
	Nissim [116]	2012	✗	◐	✓	◐	✗	✗	✗	Private
	Zhao [117]	2012	✗	✗	✗	✗	✓	✓	◐	Private
	Nissim [118]	2014	✓	✓	✗	◐	✓	✓	✗	Custom
	Zhang[119]	2015	◐	◐	✗	✗	✓	✓	✗	Private
	Nissim [120]	2016	✗	✓	✓	◐	✓	✓	◐	Custom
	Ni [121]	2016	✓	✓	✗	◐	✓	✓	◐	Private
	Chen [122]	2017	✓	✓	✗	◐	✗	✗	◐	Private
	Rashidi [66]	2017	✗	✓	✓	◐	✓	✓	✗	Drebin
	Fu [123]	2019	✓	✓	✗	✗	✓	✗	◐	Private
	Irofti [124]	2019	◐	◐	✗	◐	✗	✗	✓	DREBIN, EMBER
	Pendlebury [86]	2019	✗	✗	✓	◐	✓	✓	✓	AndroZoo
	Sharmeen [125]	2020	✓	◐	✗	◐	✓	✓	◐	Drebin, AndroZoo
	Chen [126]	2020	◐	◐	✓	✗	✓	✓	◐	MCC
	Koza [11]	2020	✓	◐	✓	◐	✓	✗	✓	Private
	Noorbehbahani [13]	2020	✓	✗	✗	◐	✓	✓	✗	AndMal17
	Li [127]	2021	✗	◐	✗	◐	✓	✗	◐	FalDroid, DREBIN, Genome
	Liang [109]	2021	✓	◐	✓	◐	✓	✓	◐	Custom

Solution: CEF-SsL

- SsL is intriguing, but its "pragmatic" benefits are still unknown
- Identifying (and quantifying) such benefits requires adopting a rigorous workflow

→ CEF-SsL: Cybersecurity Evaluation Framework for Semisupervised Learning

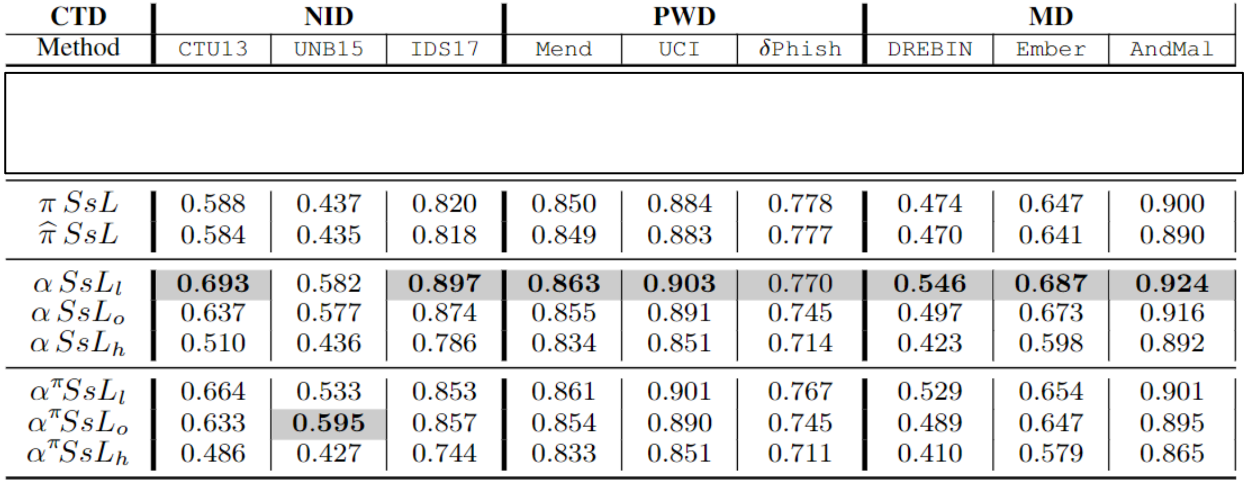

$\mathbb{D}$
$\mathcal{L}$
$\overrightarrow{ML}$
n, k
1
Prepare
$\overline{\mathbb{L}}$
$\mathbb{F}$
$\mathbb{U}$
$\mathbb{L}$
2
Run
$\overrightarrow{\varepsilon}$ one
$\overrightarrow{\mu}$ one
3
Iterate
$\overrightarrow{\varepsilon}$ nxk
$\overrightarrow{\mu}$ nxk
- (n times) keep $\mathbb{F}$, change $\mathbb{L}$
- (k times) begin anew

(re)Evaluation

- Massive evaluation on 9 existing datasets for 3 cyberthreat detection tasks:
  - Network Intrusion Detection (NID)
  - Phishing Website Detection (PWD)
  - Malware Detection (MD)

Labels range between 100 and 2400

Results (F1-score)



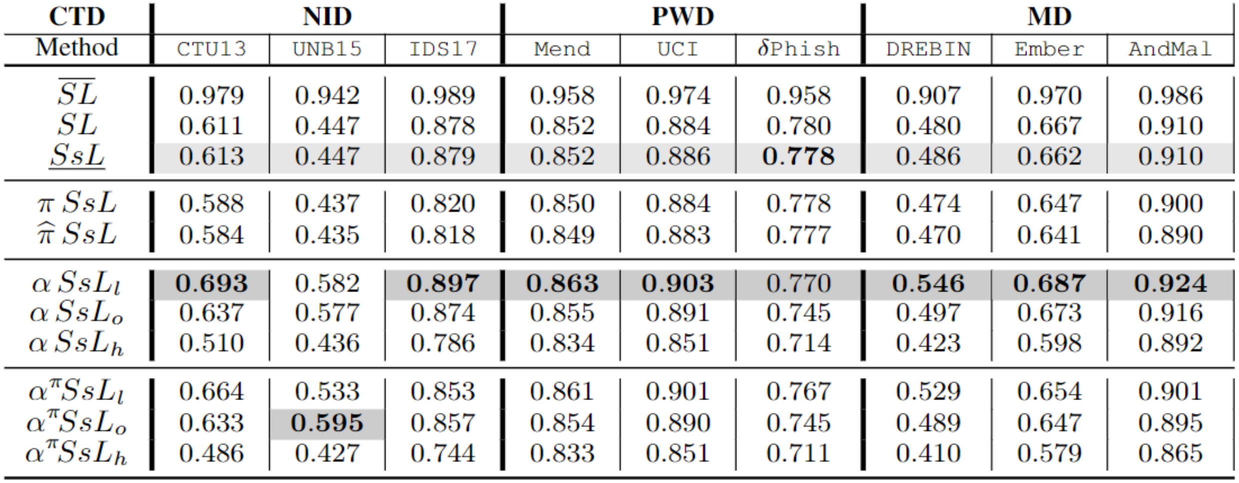
CTD	NID			PWD			MD		
Method	CTU13	UNB15	IDS17	Mend	UCI	δPhish	DREBIN	Ember	AndMal
$\pi\ SsL$	0.588	0.437	0.820	0.850	0.884	0.778	0.474	0.647	0.900
$\hat{\pi}\ SsL$	0.584	0.435	0.818	0.849	0.883	0.777	0.470	0.641	0.890
$\alpha\ SsL_l$	**0.693**	0.582	**0.897**	**0.863**	**0.903**	0.770	**0.546**	**0.687**	**0.924**
$\alpha\ SsL_o$	0.637	0.577	0.874	0.855	0.891	0.745	0.497	0.673	0.916
$\alpha\ SsL_h$	0.510	0.436	0.786	0.834	0.851	0.714	0.423	0.598	0.892
$\alpha^{\pi}SsL_l$	0.664	0.533	0.853	0.861	0.901	0.767	0.529	0.654	0.901
$\alpha^{\pi}SsL_o$	0.633	**0.595**	0.857	0.854	0.890	0.745	0.489	0.647	0.895
$\alpha^{\pi}SsL_h$	0.486	0.427	0.744	0.833	0.851	0.711	0.410	0.579	0.865

(re)Evaluation

- Massive evaluation on 9 existing datasets for 3 cyberthreat detection tasks:
  - Network Intrusion Detection (NID)
  - Phishing Website Detection (PWD)
  - Malware Detection (MD)

Labels range between 100 and 2400

Results (F1-score)



CTD	NID			PWD			MD		
Method	CTU13	UNB15	IDS17	Mend	UCI	δPhish	DREBIN	Ember	AndMal
$\overline{SL}$	0.979	0.942	0.989	0.958	0.974	0.958	0.907	0.970	0.986
SL	0.611	0.447	0.878	0.852	0.884	0.780	0.480	0.667	0.910
$\underline{SsL}$	0.613	0.447	0.879	0.852	0.886	**0.778**	0.486	0.662	0.910
$\pi\, SsL$	0.588	0.437	0.820	0.850	0.884	0.778	0.474	0.647	0.900
$\widehat{\pi}\, SsL$	0.584	0.435	0.818	0.849	0.883	0.777	0.470	0.641	0.890
$\alpha\, SsL_l$	**0.693**	0.582	**0.897**	**0.863**	**0.903**	0.770	**0.546**	**0.687**	**0.924**
$\alpha\, SsL_o$	0.637	0.577	0.874	0.855	0.891	0.745	0.497	0.673	0.916
$\alpha\, SsL_h$	0.510	0.436	0.786	0.834	0.851	0.714	0.423	0.598	0.892
$\alpha^{\pi} SsL_l$	0.664	0.533	0.853	0.861	0.901	0.767	0.529	0.654	0.901
$\alpha^{\pi} SsL_o$	0.633	**0.595**	0.857	0.854	0.890	0.745	0.489	0.647	0.895
$\alpha^{\pi} SsL_h$	0.486	0.427	0.744	0.833	0.851	0.711	0.410	0.579	0.865

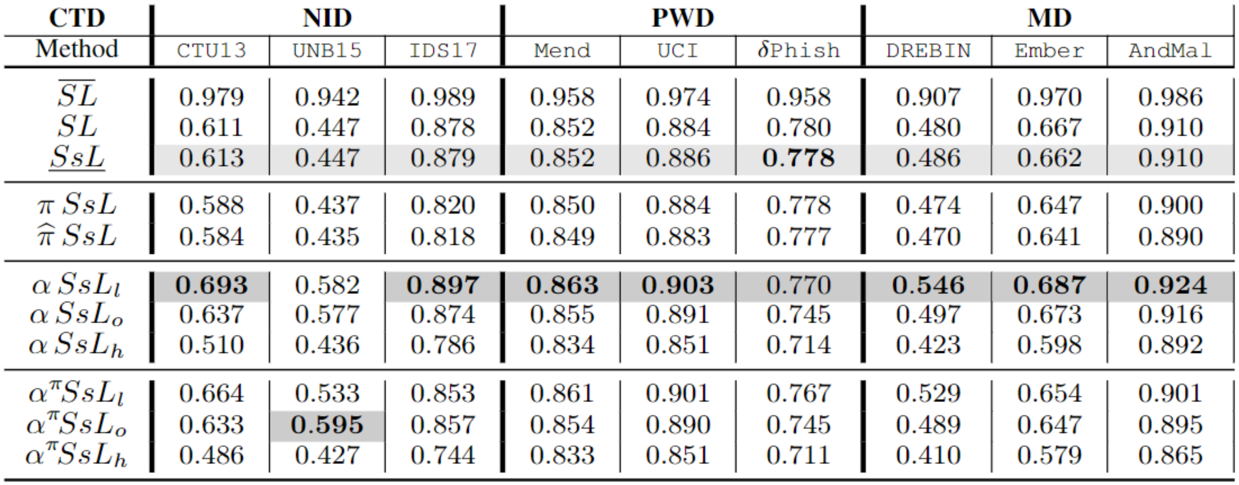
Is SsL truly advantageous?

(re)Evaluation

- Massive evaluation on 9 existing datasets for 3 cyberthreat detection tasks:
  - Network Intrusion Detection (NID)
  - Phishing Website Detection (PWD)
  - Malware Detection (MD)

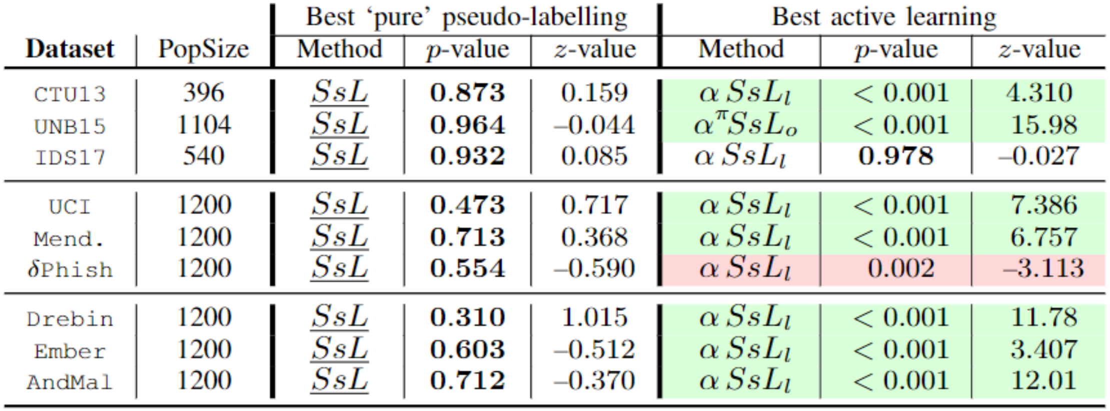
Labels range between 100 and 2400

Results (F1-score)



CTD	NID			PWD			MD		
Method	CTU13	UNB15	IDS17	Mend	UCI	δPhish	DREBIN	Ember	AndMal
$\overline{SL}$	0.979	0.942	0.989	0.958	0.974	0.958	0.907	0.970	0.986
SL	0.611	0.447	0.878	0.852	0.884	0.780	0.480	0.667	0.910
$\underline{SsL}$	0.613	0.447	0.879	0.852	0.886	**0.778**	0.486	0.662	0.910
$\pi\, SsL$	0.588	0.437	0.820	0.850	0.884	0.778	0.474	0.647	0.900
$\widehat{\pi}\, SsL$	0.584	0.435	0.818	0.849	0.883	0.777	0.470	0.641	0.890
$\alpha\, SsL_l$	**0.693**	0.582	**0.897**	**0.863**	**0.903**	0.770	**0.546**	**0.687**	**0.924**
$\alpha\, SsL_o$	0.637	0.577	0.874	0.855	0.891	0.745	0.497	0.673	0.916
$\alpha\, SsL_h$	0.510	0.436	0.786	0.834	0.851	0.714	0.423	0.598	0.892
$\alpha^{\pi} SsL_l$	0.664	0.533	0.853	0.861	0.901	0.767	0.529	0.654	0.901
$\alpha^{\pi} SsL_o$	0.633	**0.595**	0.857	0.854	0.890	0.745	0.489	0.647	0.895
$\alpha^{\pi} SsL_h$	0.486	0.427	0.744	0.833	0.851	0.711	0.410	0.579	0.865

Statistical Validation



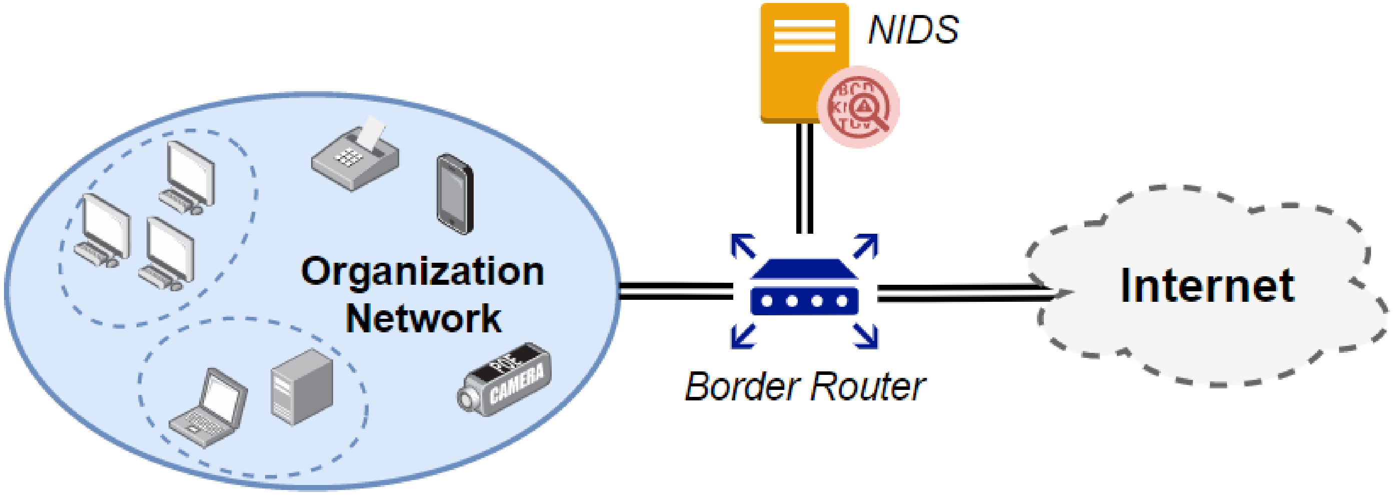
		Best 'pure' pseudo-labelling			Best active learning		
Dataset	PopSize	Method	p-value	z-value	Method	p-value	z-value
CTU13	396	$\underline{SsL}$	**0.873**	0.159	$\alpha\, SsL_l$	< 0.001	4.310
UNB15	1104	$\underline{SsL}$	**0.964**	–0.044	$\alpha^{\pi} SsL_o$	< 0.001	15.98
IDS17	540	$\underline{SsL}$	**0.932**	0.085	$\alpha\, SsL_l$	**0.978**	–0.027
UCI	1200	$\underline{SsL}$	**0.473**	0.717	$\alpha\, SsL_l$	< 0.001	7.386
Mend.	1200	$\underline{SsL}$	**0.713**	0.368	$\alpha\, SsL_l$	< 0.001	6.757
δPhish	1200	$\underline{SsL}$	**0.554**	–0.590	$\alpha\, SsL_l$	0.002	–3.113
Drebin	1200	$\underline{SsL}$	**0.310**	1.015	$\alpha\, SsL_l$	< 0.001	11.78
Ember	1200	$\underline{SsL}$	**0.603**	–0.512	$\alpha\, SsL_l$	< 0.001	3.407
AndMal	1200	$\underline{SsL}$	**0.712**	–0.370	$\alpha\, SsL_l$	< 0.001	12.01

UNIVERSITÄT LIECHTENSTEIN

Improving Machine Learning in Network Intrusion Detection

Problem Statement

- Most organizations adopt Network Intrusion Detection Systems (NIDS)
- Such NIDS are starting to actively leverage Machine Learning (ML-NIDS)

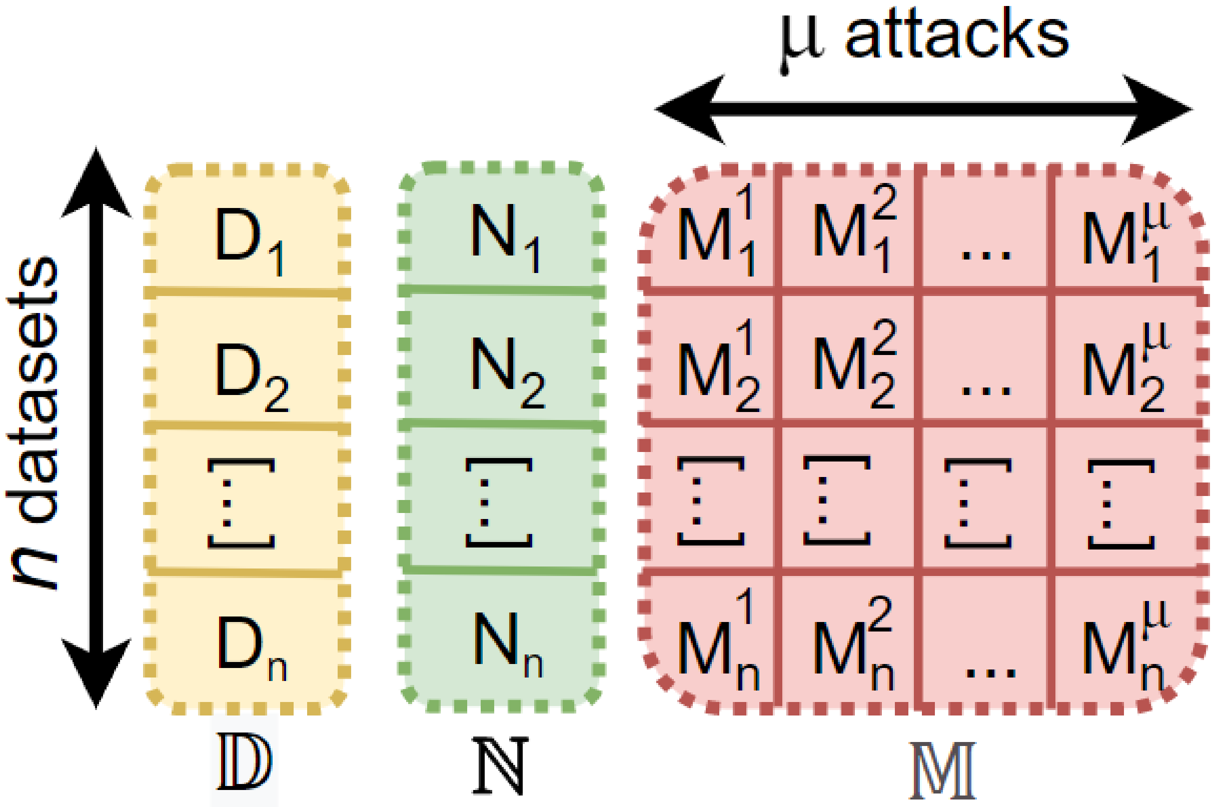


- **Problem: every network environment is *unique***
  - This characteristic conflicts with the “iid” assumption, which is fundamental for ML
    - iid: independent and identically distributed random variables
  - Training data must be collected *from and for* each network monitored by ML-NIDS

Intuition: Cross-evaluation of ML-NIDS

- ○ It is true that every network is unique…
- ○ … but (some) *malicious* events are malicious *everywhere* and *everytime*

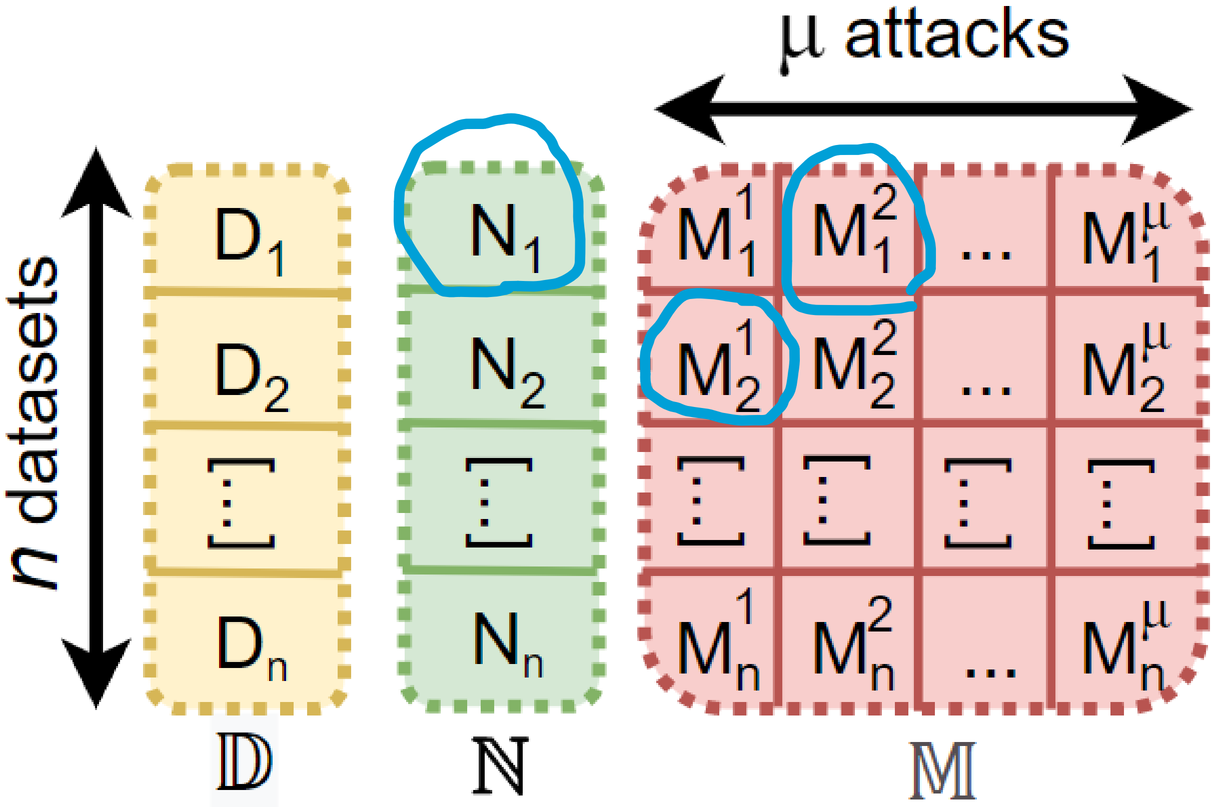
→ Why not using malicious samples taken from *different* networks to "augment" the data used for training/testing my ML models?



Intuition: Cross-evaluation of ML-NIDS

- It is true that every network is unique…
- … but (some) *malicious* events are malicious *everywhere* and *everytime*

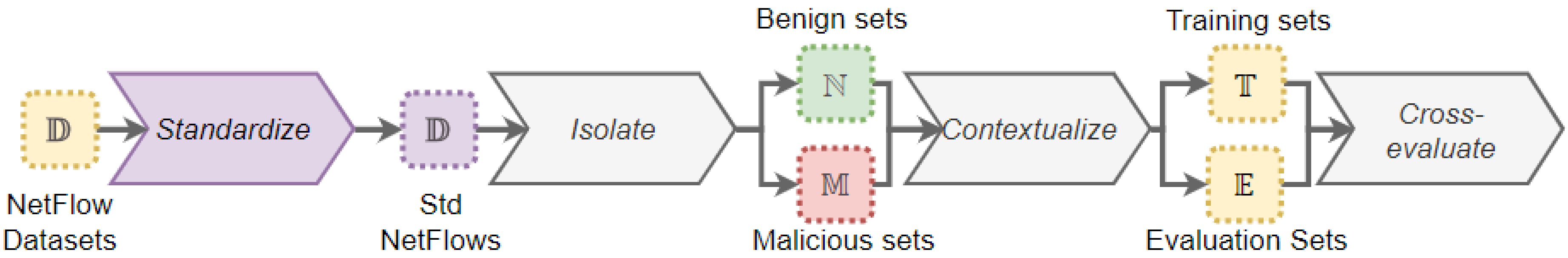
→ Why not using malicious samples taken from *different* networks to "augment" the data used for training/testing my ML models?



Solution: XeNIDS

- The idea is intriguing, but *applying* it in practice is difficult
  - Adversarial poisoning
  - Incompatible networks
  - False-sense of security
  - Performance Decrease

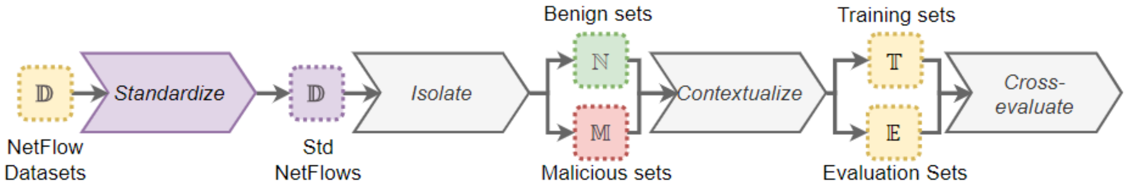
→ XeNIDS – framework for the Cross-evaluation of Network Intrusion Detection Systems

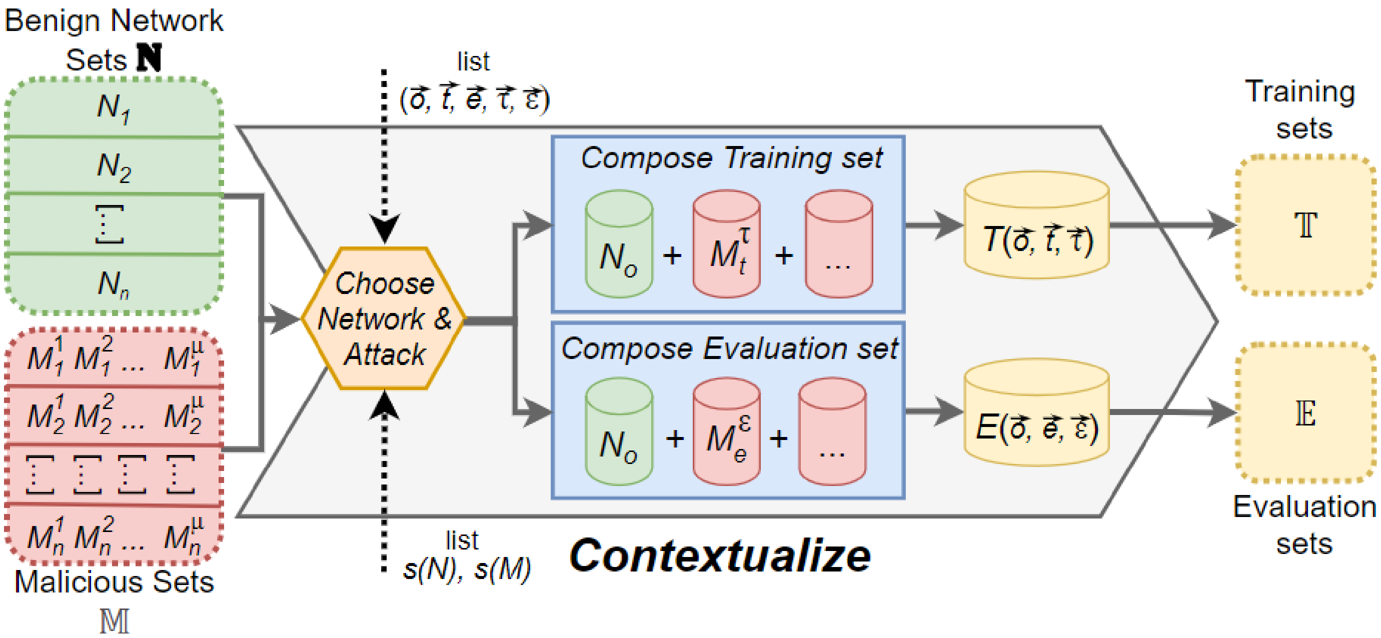


Solution: XeNIDS

- The idea is intriguing, but *applying* it in practice is difficult
  - Adversarial poisoning
  - Incompatible networks
  - False-sense of security
  - Performance Decrease

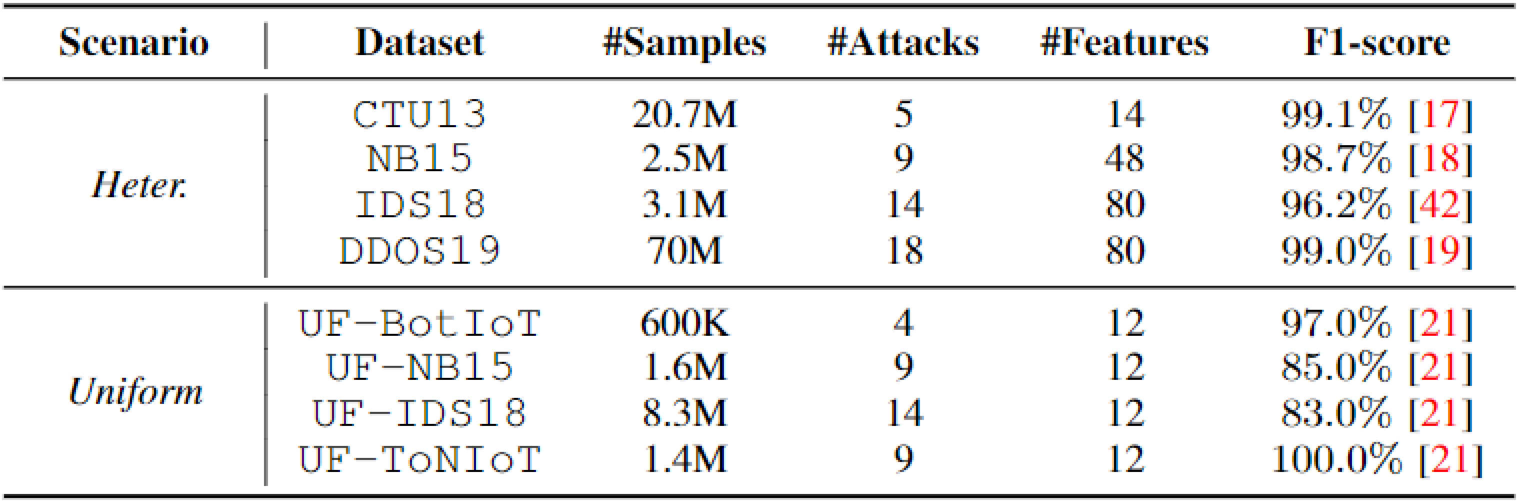
→ XeNIDS – framework for the Cross-evaluation of Network Intrusion Detection Systems




Benign Network Sets $\mathbb{N}$
N_1
N_2
$\vdots$
N_n
$M_1^1\ M_1^2 \ldots\ M_1^\mu$
$M_2^1\ M_2^2 \ldots\ M_2^\mu$
$\vdots\ \vdots\ \vdots\ \vdots$
$M_n^1\ M_n^2 \ldots\ M_n^\mu$
Malicious Sets
$\mathbb{M}$
list
$(\vec{o}, \vec{t}, \vec{e}, \vec{\tau}, \vec{\varepsilon})$
Choose Network & Attack
list
$s(N)$, $s(M)$
Compose Training set
N_o + M_t^τ + ...
Compose Evaluation set
N_o + M_e^ε + ...
$T(\vec{o}, \vec{t}, \vec{\tau})$
$E(\vec{o}, \vec{e}, \vec{\varepsilon})$
Contextualize
Training sets
$\mathbb{T}$
$\mathbb{E}$
Evaluation sets

Evaluation

- Massive evaluation of XeNIDS on 8 datasets

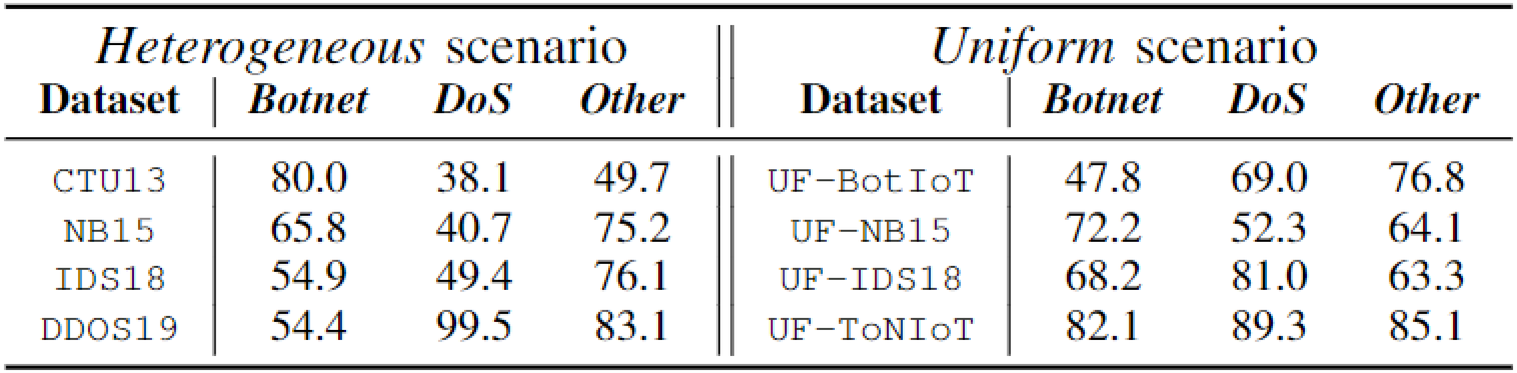


Scenario	Dataset	#Samples	#Attacks	#Features	F1-score
Heter.	CTU13	20.7M	5	14	99.1% [17]
	NB15	2.5M	9	48	98.7% [18]
	IDS18	3.1M	14	80	96.2% [42]
	DDOS19	70M	18	80	99.0% [19]
Uniform	UF-BotIoT	600K	4	12	97.0% [21]
	UF-NB15	1.6M	9	12	85.0% [21]
	UF-IDS18	8.3M	14	12	83.0% [21]
	UF-ToNIoT	1.4M	9	12	100.0% [21]

- XeNIDS can be used for:
  - Assessing how an existing ML-NIDS fares against “unknown” attacks; and
  - Increasing the robustness of an existing ML-NIDS against such “unknown” attacks

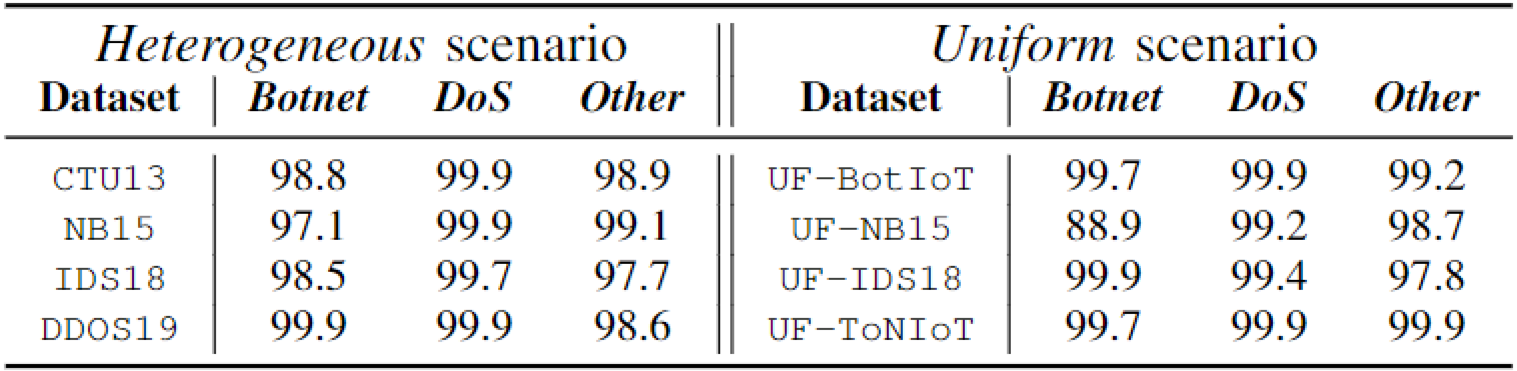
Results

- Baseline performance against unknown attacks (F1-score)



Heterogeneous scenario				*Uniform* scenario			
Dataset	***Botnet***	***DoS***	***Other***	**Dataset**	***Botnet***	***DoS***	***Other***
CTU13	80.0	38.1	49.7	UF-BotIoT	47.8	69.0	76.8
NB15	65.8	40.7	75.2	UF-NB15	72.2	52.3	64.1
IDS18	54.9	49.4	76.1	UF-IDS18	68.2	81.0	63.3
DDOS19	54.4	99.5	83.1	UF-ToNIoT	82.1	89.3	85.1

- Enhanced performance against unknown attacks (F1-score)

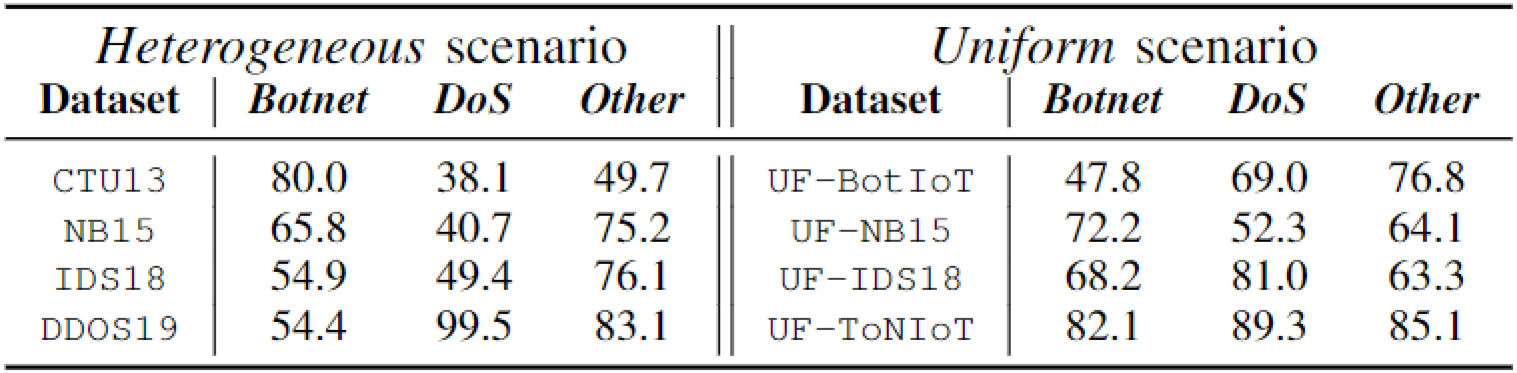


Heterogeneous scenario				*Uniform* scenario			
Dataset	***Botnet***	***DoS***	***Other***	**Dataset**	***Botnet***	***DoS***	***Other***
CTU13	98.8	99.9	98.9	UF-BotIoT	99.7	99.9	99.2
NB15	97.1	99.9	99.1	UF-NB15	88.9	99.2	98.7
IDS18	98.5	99.7	97.7	UF-IDS18	99.9	99.4	97.8
DDOS19	99.9	99.9	98.6	UF-ToNIoT	99.7	99.9	99.9

Results

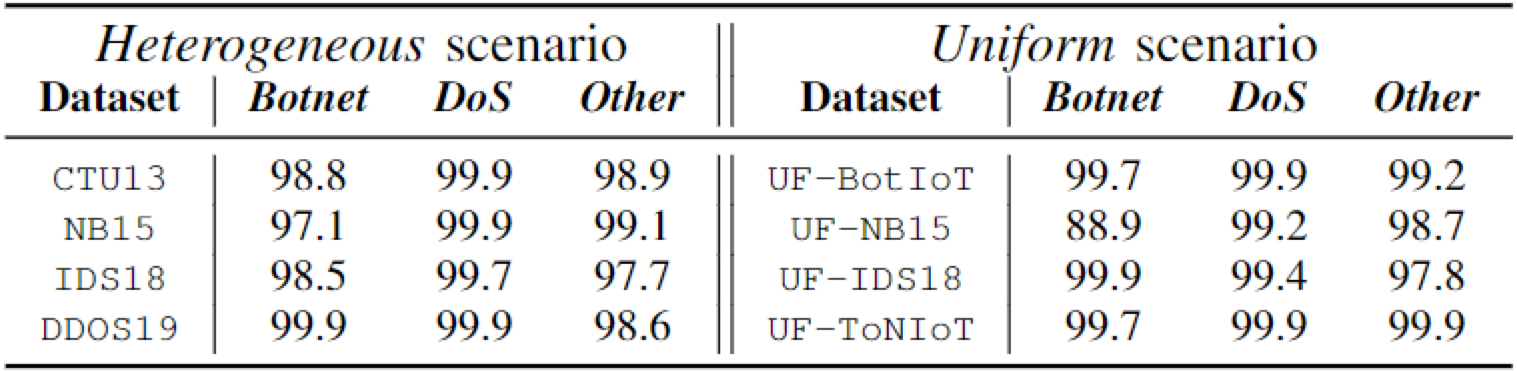
Advantage?
It's all free

- Baseline performance against unknown attacks (F1-score)



Heterogeneous scenario				*Uniform* scenario			
Dataset	***Botnet***	***DoS***	***Other***	**Dataset**	***Botnet***	***DoS***	***Other***
CTU13	80.0	38.1	49.7	UF-BotIoT	47.8	69.0	76.8
NB15	65.8	40.7	75.2	UF-NB15	72.2	52.3	64.1
IDS18	54.9	49.4	76.1	UF-IDS18	68.2	81.0	63.3
DDOS19	54.4	99.5	83.1	UF-ToNIoT	82.1	89.3	85.1

- Enhanced performance against unknown attacks (F1-score)



Heterogeneous scenario				*Uniform* scenario			
Dataset	***Botnet***	***DoS***	***Other***	**Dataset**	***Botnet***	***DoS***	***Other***
CTU13	98.8	99.9	98.9	UF-BotIoT	99.7	99.9	99.2
NB15	97.1	99.9	99.1	UF-NB15	88.9	99.2	98.7
IDS18	98.5	99.7	97.7	UF-IDS18	99.9	99.4	97.8
DDOS19	99.9	99.9	98.6	UF-ToNIoT	99.7	99.9	99.9

CAUTION!
Always analyze the results!

The security of Machine Learning-based Phishing Website Detectors

Current Landscape of Phishing

- Phishing attacks are continuously increasing
- Current detection methods still rely on *blacklists* of malicious URLs
  - These detection techniques can be evaded easily by “squatting” phishing websites!

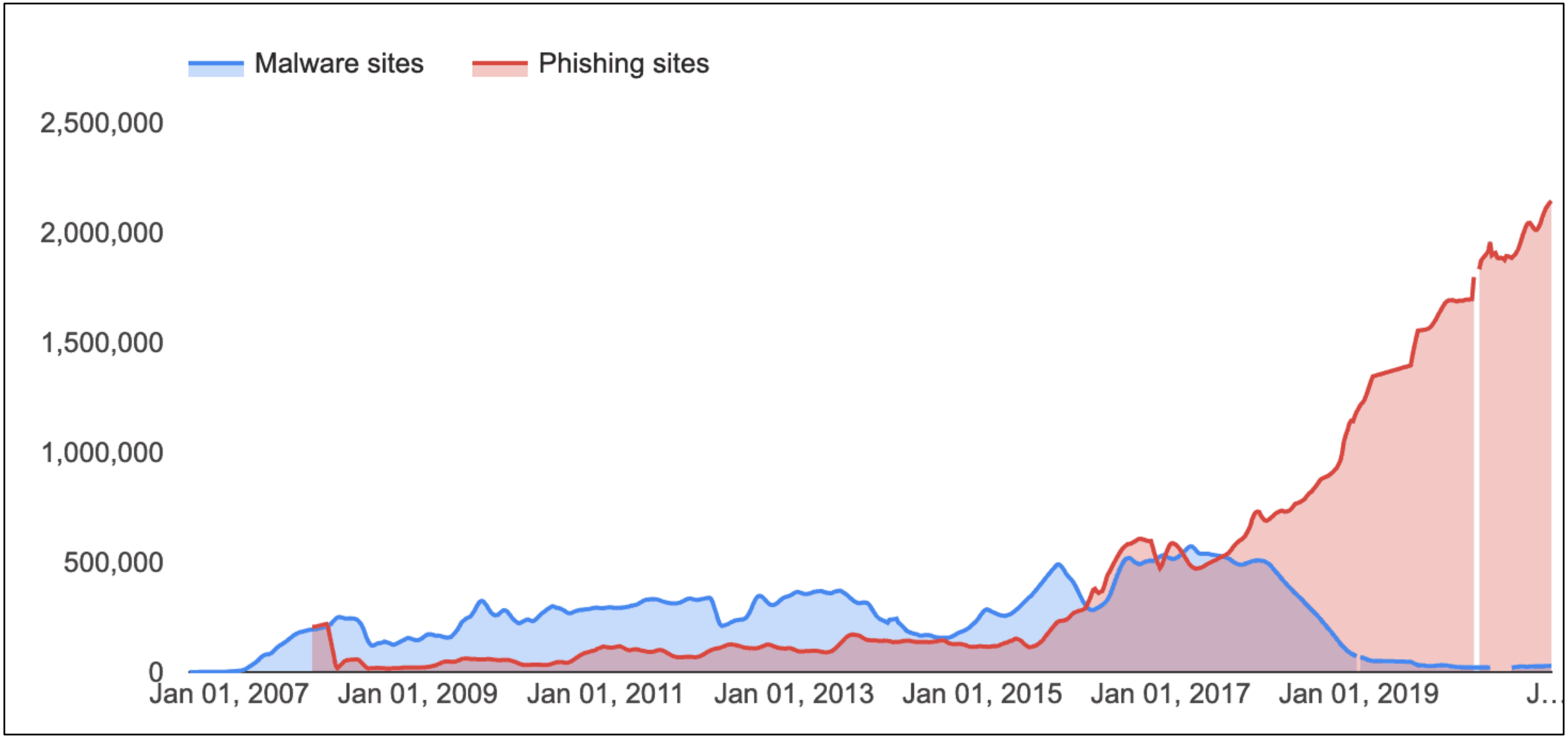
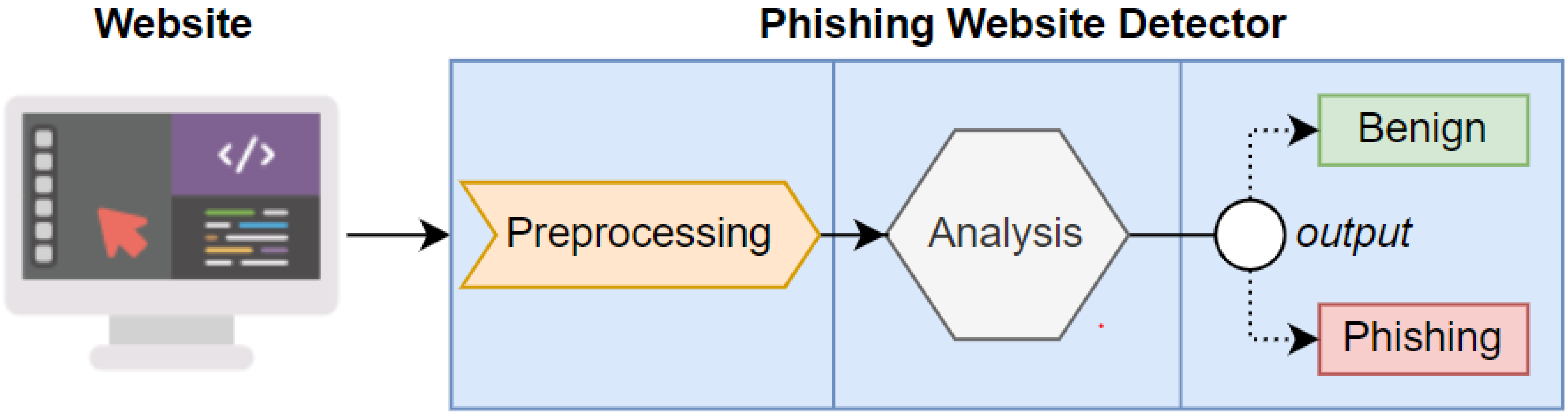


Image source: https://www.tessian.com/blog/phishing-statistics-2020/

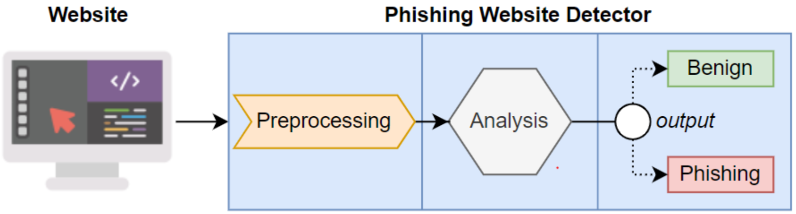
Current Landscape of Phishing – Countermeasures

- Countering such simple (but effective) strategies can be done via *data-driven* methods

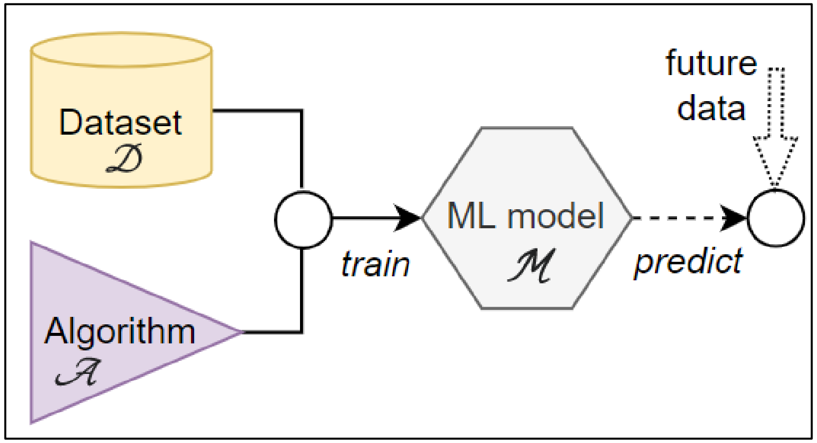


Current Landscape of Phishing – Countermeasures (ML)

- Countering such simple (but effective) strategies can be done via *data-driven* methods



- Such methods (obviously ☺) include (also) Machine Learning techniques:



- Machine Learning-based Phishing Website Detectors (ML-PWD) are very effective! [1]
  - Even popular products and web-browsers (e.g., Google Chrome) use them! [2]

[1]: Tian, Ke, et al. "Needle in a haystack: Tracking down elite phishing domains in the wild." Internet Measurement Conference 2018.
[2]: El Kouari, Oumaima, Hafssa Benaboud, and Saiida Lazaar. "Using machine learning to deal with Phishing and Spam Detection: An overview." Proceedings of the 3rd International Conference on Networking, Information Systems & Security. 2020.

Problem Statement: Adversarial Attacks against ML

- ML-PWD are good but…
- …the detection of ML methods *can* be bypassed via (adversarial) *evasion* attacks!
- Adversarial Attacks exploit a perturbation, ε, that induces a ML model, $\mathcal{M}$, to misclassify a given input, x, by producing an incorrect output (y_x^ε instead of y_x)

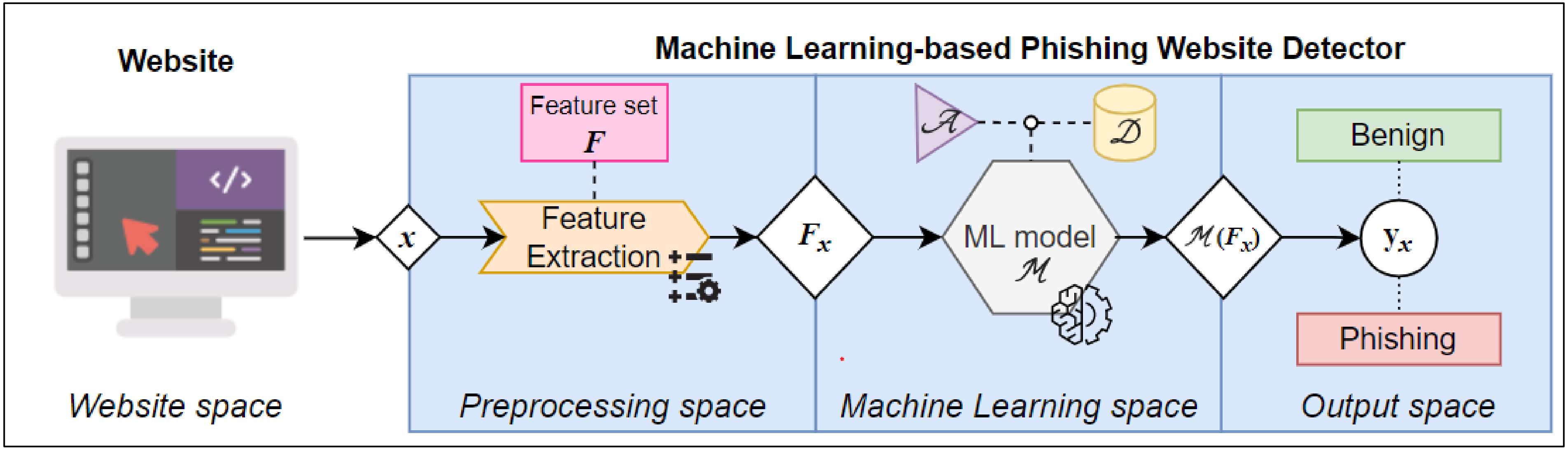
$$\text{find } \varepsilon \text{ s.t. } \mathcal{M}(F_x) = y_x^\varepsilon \neq y_x$$

Problem Statement: Adversarial Attacks against ML-PWD

- ML-PWD are good but…
- …the detection of ML methods *can* be bypassed via (adversarial) *evasion* attacks!
- Adversarial Attacks exploit a perturbation, ε, that induces a ML model, $\mathcal{M}$, to misclassify a given input, x, by producing an incorrect output (y_x^ε instead of y_x)

$$\text{find } \varepsilon \text{ s.t. } \mathcal{M}(F_x) = y_x^\varepsilon \neq y_x$$

- In the context of a ML-PWD, such ε can be introduced in three 'spaces':



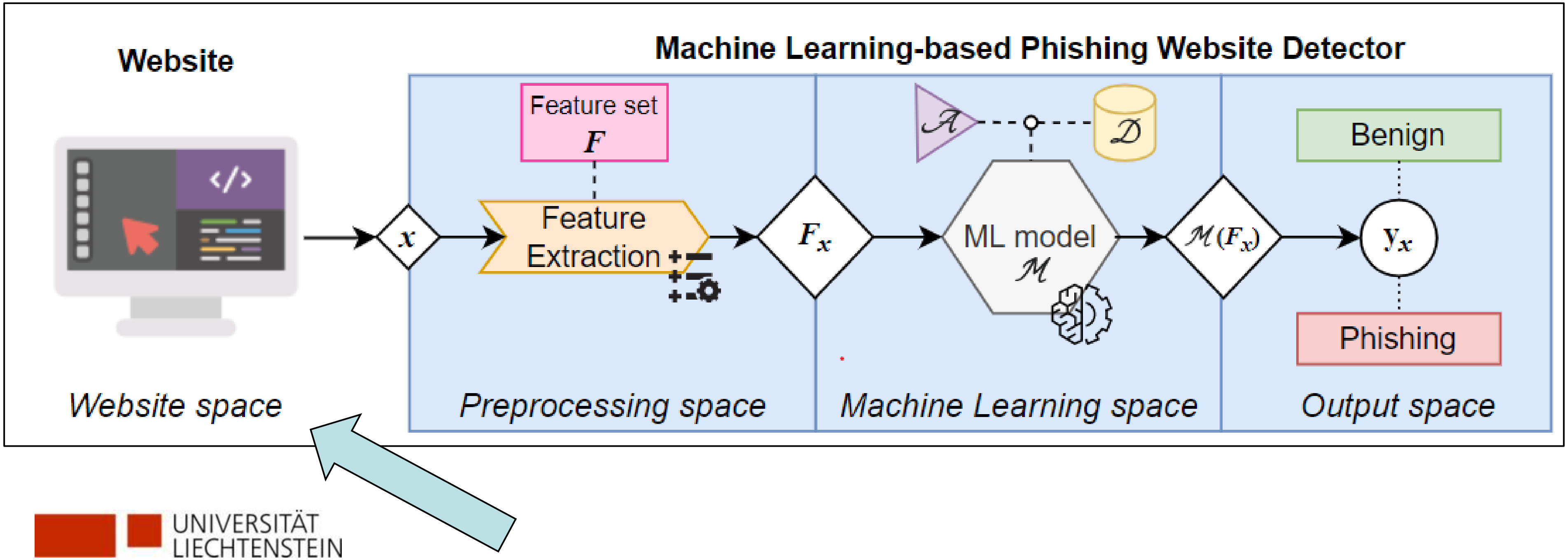
UNIVERSITÄT LIECHTENSTEIN

Problem Statement: Adversarial Attacks against ML-PWD

- ML-PWD are good but…
- …the detection of ML methods *can* be bypassed via (adversarial) *evasion* attacks!
- Adversarial Attacks exploit a perturbation, ε, that induces a ML model, $\mathcal{M}$, to misclassify a given input, x, by producing an incorrect output (y_x^ε instead of y_x)

$$\text{find } \varepsilon \text{ s.t. } \mathcal{M}(F_x) = y_x^\varepsilon \neq y_x$$

- In the context of a ML-PWD, such ε can be introduced in three 'spaces':

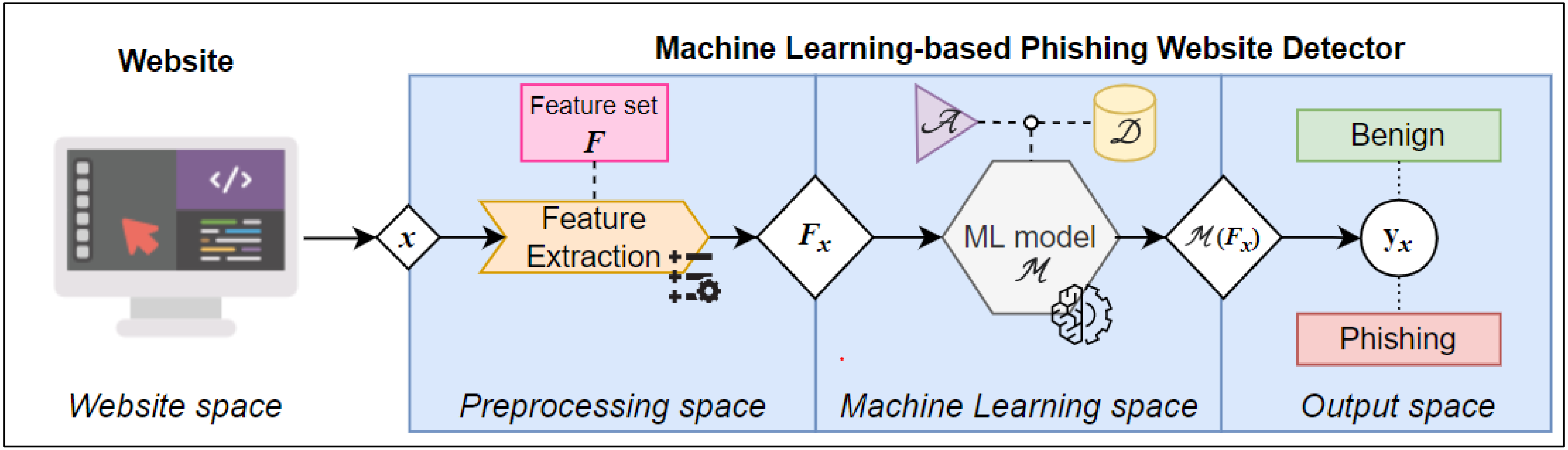


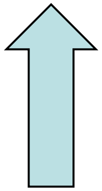
Problem Statement: Adversarial Attacks against ML-PWD

- ML-PWD are good but…
- …the detection of ML methods *can* be bypassed via (adversarial) *evasion* attacks!
- Adversarial Attacks exploit a perturbation, ε, that induces a ML model, $\mathcal{M}$, to misclassify a given input, x, by producing an incorrect output (y_x^ε instead of y_x)

$$\text{find } \varepsilon \text{ s.t. } \mathcal{M}(F_x) = y_x^\varepsilon \neq y_x$$

- In the context of a ML-PWD, such ε can be introduced in three 'spaces':





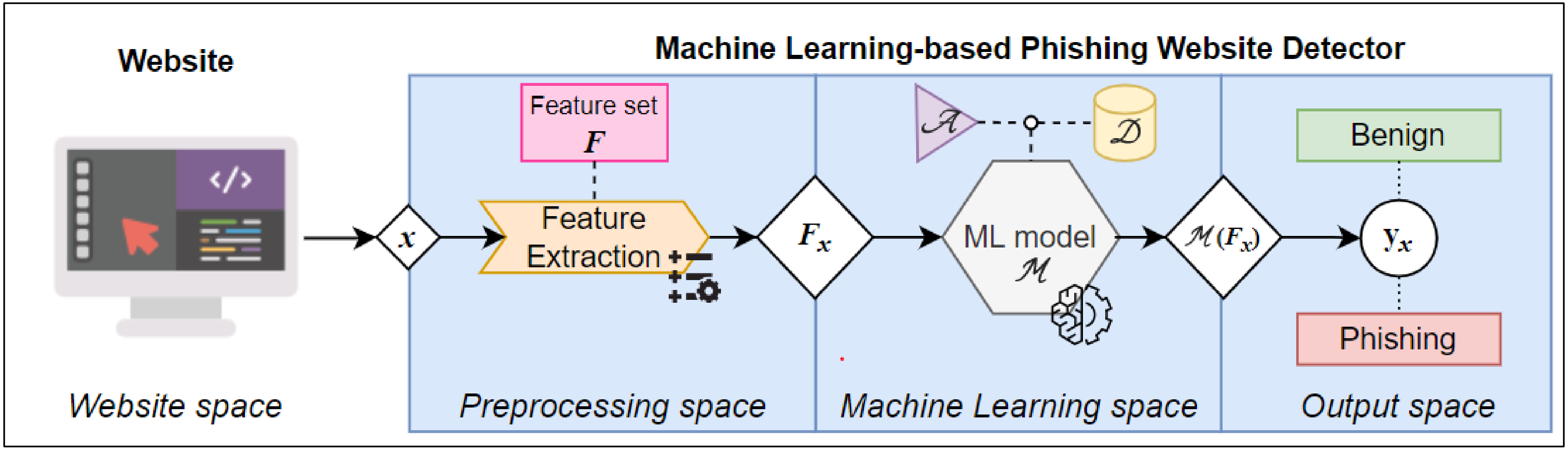
Giovanni Apruzzese, *PhD*
giovanni.apruzzese@uni.li

Problem Statement: Adversarial Attacks against ML-PWD

- ML-PWD are good but…
- …the detection of ML methods *can* be bypassed via (adversarial) *evasion* attacks!
- Adversarial Attacks exploit a perturbation, ε, that induces a ML model, $\mathcal{M}$, to misclassify a given input, x, by producing an incorrect output (y_x^ε instead of y_x)

$$\text{find } \varepsilon \text{ s.t. } \mathcal{M}(F_x) = y_x^\varepsilon \neq y_x$$

- In the context of a ML-PWD, such ε can be introduced in three 'spaces':

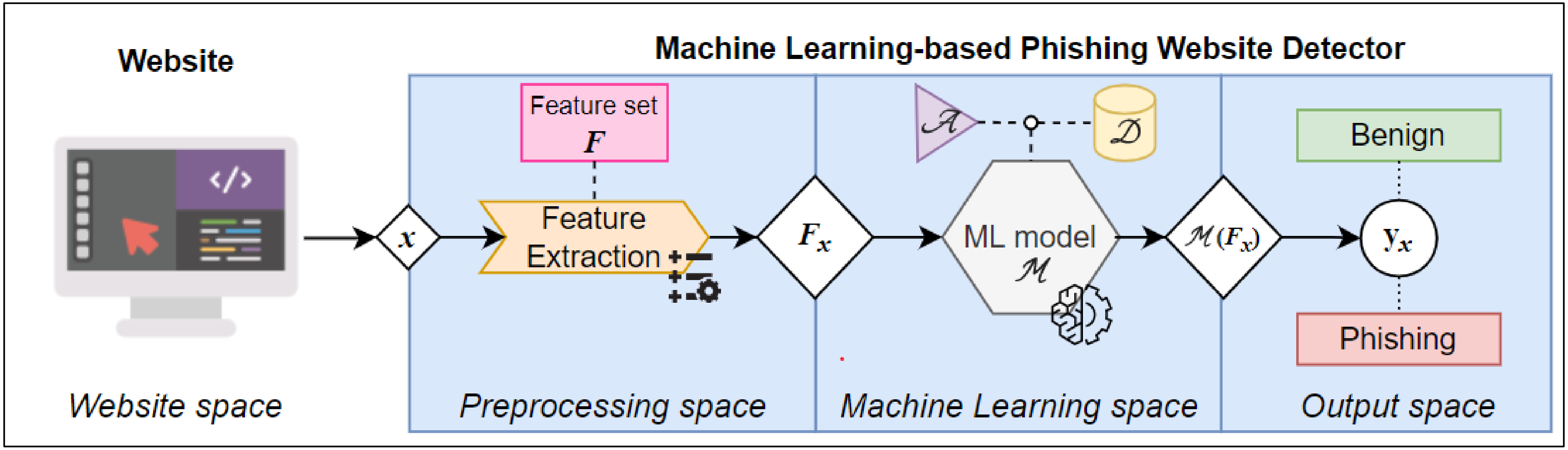


Problem Statement: Adversarial Attacks against ML-PWD

- ML-PWD are good but…
- …the detection of ML methods *can* be bypassed via (adversarial) *evasion* attacks!
- Adversarial Attacks exploit a perturbation, ε, that induces a ML model, $\mathcal{M}$, to misclassify a given input, x, by producing an incorrect output (y_x^ε instead of y_x)

$$\text{find } \varepsilon \text{ s.t. } \mathcal{M}(F_x) = y_x^\varepsilon \neq y_x$$

- In the context of a ML-PWD, such ε can be introduced in three 'spaces':

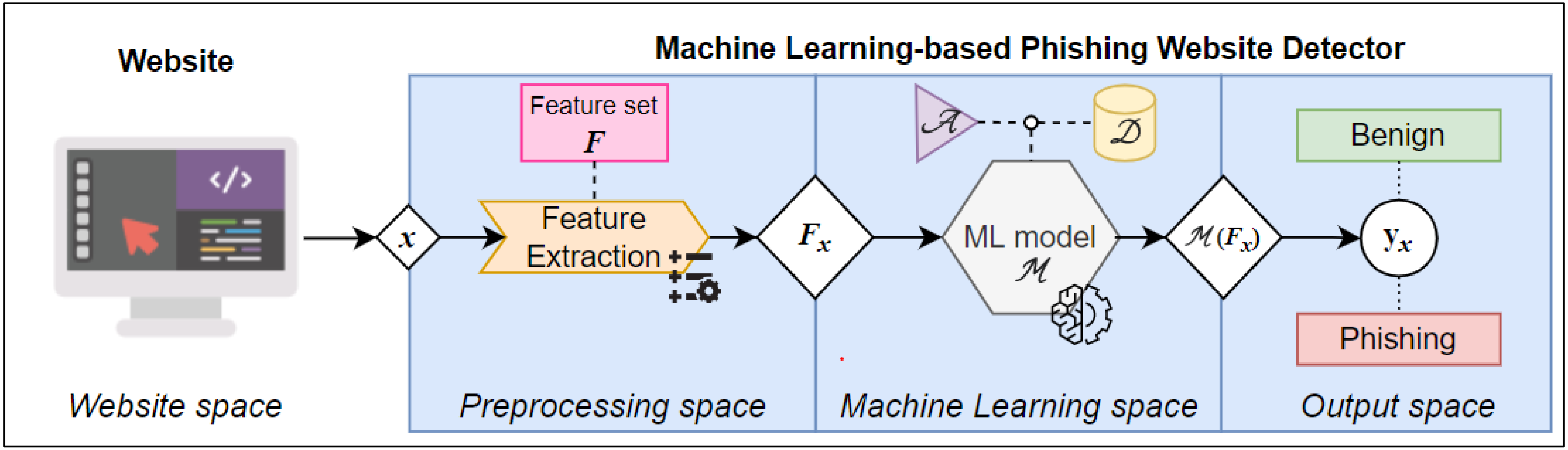


Problem Statement: Adversarial Attacks against ML-PWD

- ML-PWD are good but…
- …the detection of ML methods *can* be bypassed via (adversarial) *evasion* attacks!
- Adversarial Attacks exploit a perturbation, ε, that induces a ML model, $\mathcal{M}$, to misclassify a given input, x, by producing an incorrect output (y_x^ε instead of y_x)

$$\text{find } \varepsilon \text{ s.t. } \mathcal{M}(F_x) = y_x^\varepsilon \neq y_x$$

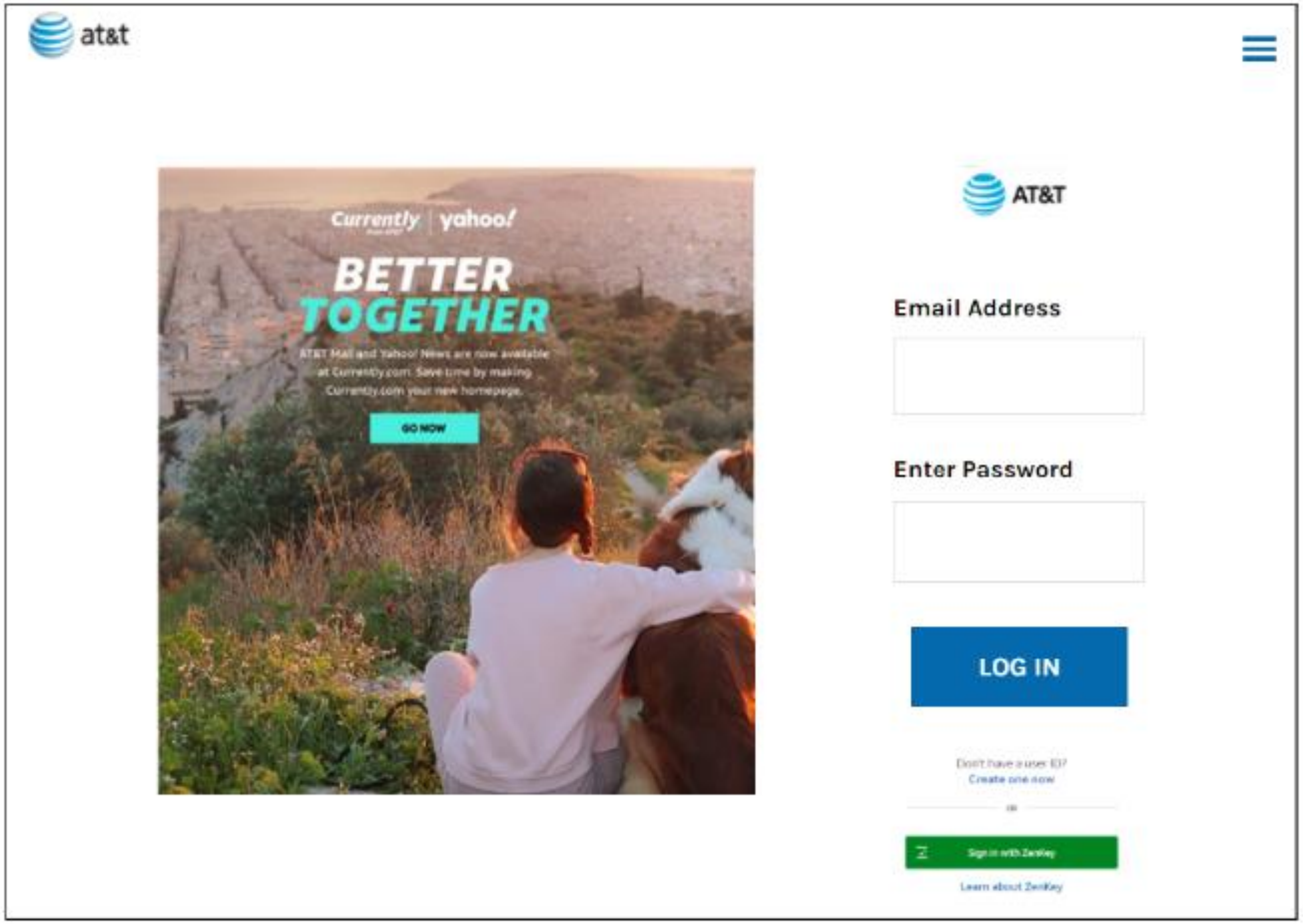
- In the context of a ML-PWD, such ε can be introduced in three "spaces":



Question: Which "space" do you think an *attacker* is **most likely** to use?

Website-space Perturbations – In practice (original example)

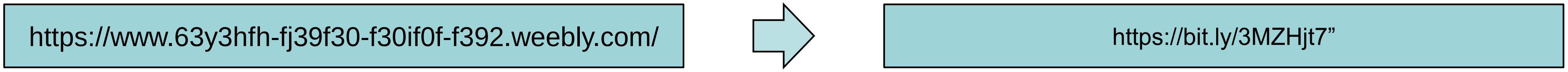
Figure 4: An exemplary (and true) Phishing website, whose URL is https://www.63y3hfh-fj39f30-f30if0f-f392.weebly.com/.



AT&T Mail and Yahoo! News are now available at Currently.com. Save time by making Currently.com your new homepage.

Don't have a user ID?
Create one now

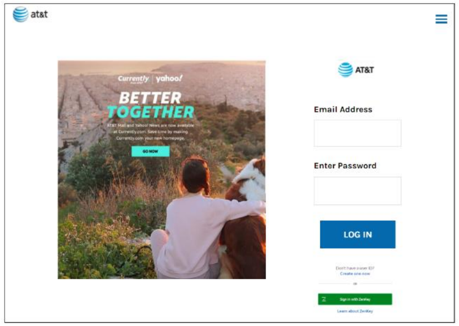
Website-space Perturbations – In practice (changing the URL)

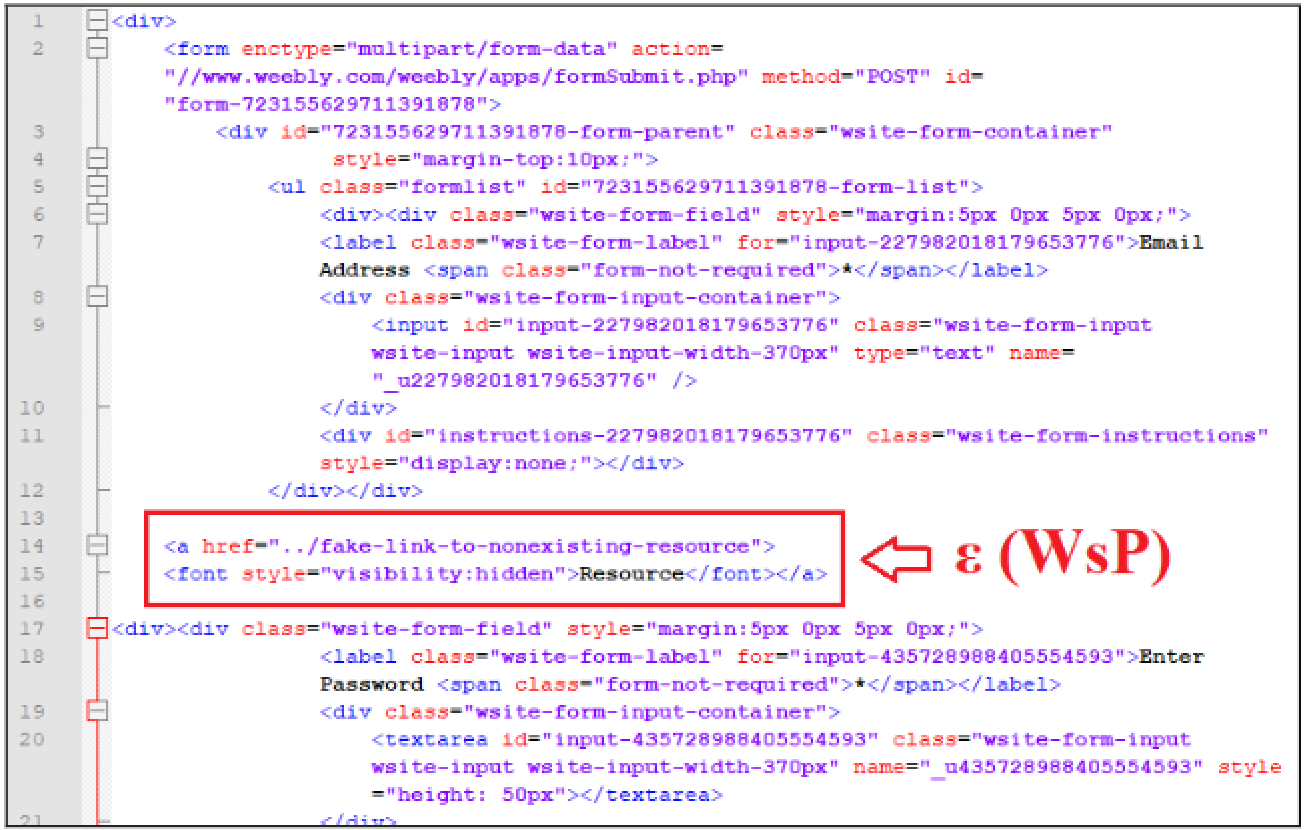


https://www.63y3hfh-fj39f30-f30if0f-f392.weebly.com/

https://bit.ly/3MZHjt7”

Website-space Perturbations – In practice (changing the HTML)



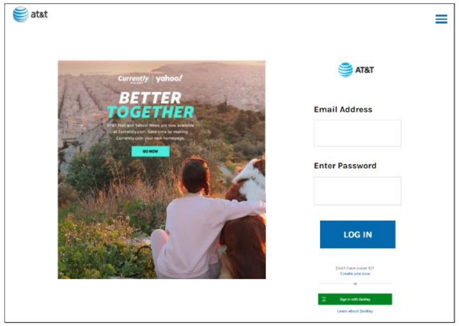


```
<div>
    <form enctype="multipart/form-data" action=
    "//www.weebly.com/weebly/apps/formSubmit.php" method="POST" id=
    "form-723155629711391878">
        <div id="723155629711391878-form-parent" class="wsite-form-container"
                  style="margin-top:10px;">
            <ul class="formlist" id="723155629711391878-form-list">
                <div><div class="wsite-form-field" style="margin:5px 0px 5px 0px;">
                <label class="wsite-form-label" for="input-227982018179653776">Email
                Address <span class="form-not-required">*</span></label>
                <div class="wsite-form-input-container">
                    <input id="input-227982018179653776" class="wsite-form-input
                    wsite-input wsite-input-width-370px" type="text" name=
                    "_u227982018179653776" />
                </div>
                <div id="instructions-227982018179653776" class="wsite-form-instructions"
                style="display:none;"></div>
            </div></div>

    <a href="../fake-link-to-nonexisting-resource">
    <font style="visibility:hidden">Resource</font></a>

<div><div class="wsite-form-field" style="margin:5px 0px 5px 0px;">
                <label class="wsite-form-label" for="input-435728988405554593">Enter
                Password <span class="form-not-required">*</span></label>
                <div class="wsite-form-input-container">
                    <textarea id="input-435728988405554593" class="wsite-form-input
                    wsite-input wsite-input-width-370px" name="_u435728988405554593" style
                    ="height: 50px"></textarea>
                </div>
```

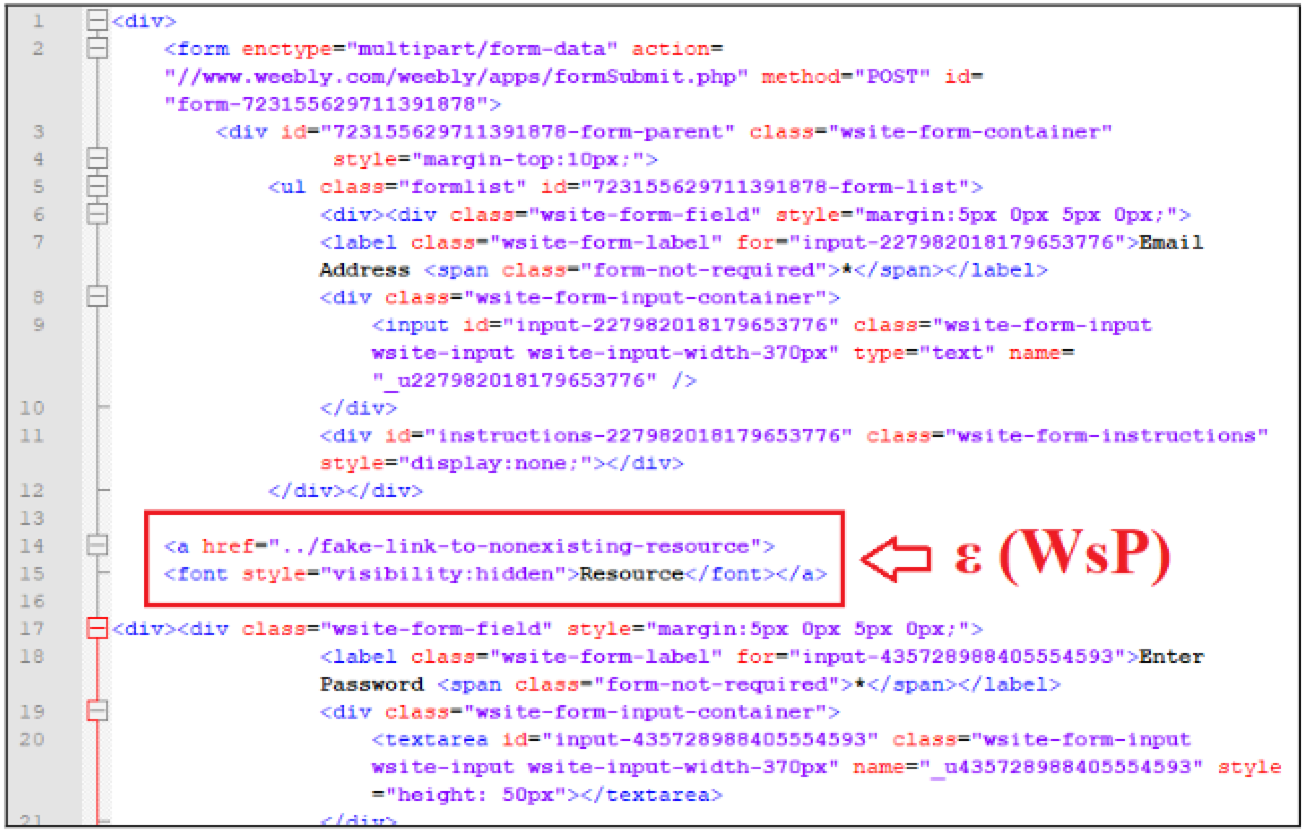
UNIVERSITÄT
LIECHTENSTEIN

Website-space Perturbations – In practice (change URL + HTML)

https://www.63y3hfh-fj39f30-f30if0f-f392.weebly.com/ → https://bit.ly/3MZHjt7”



→

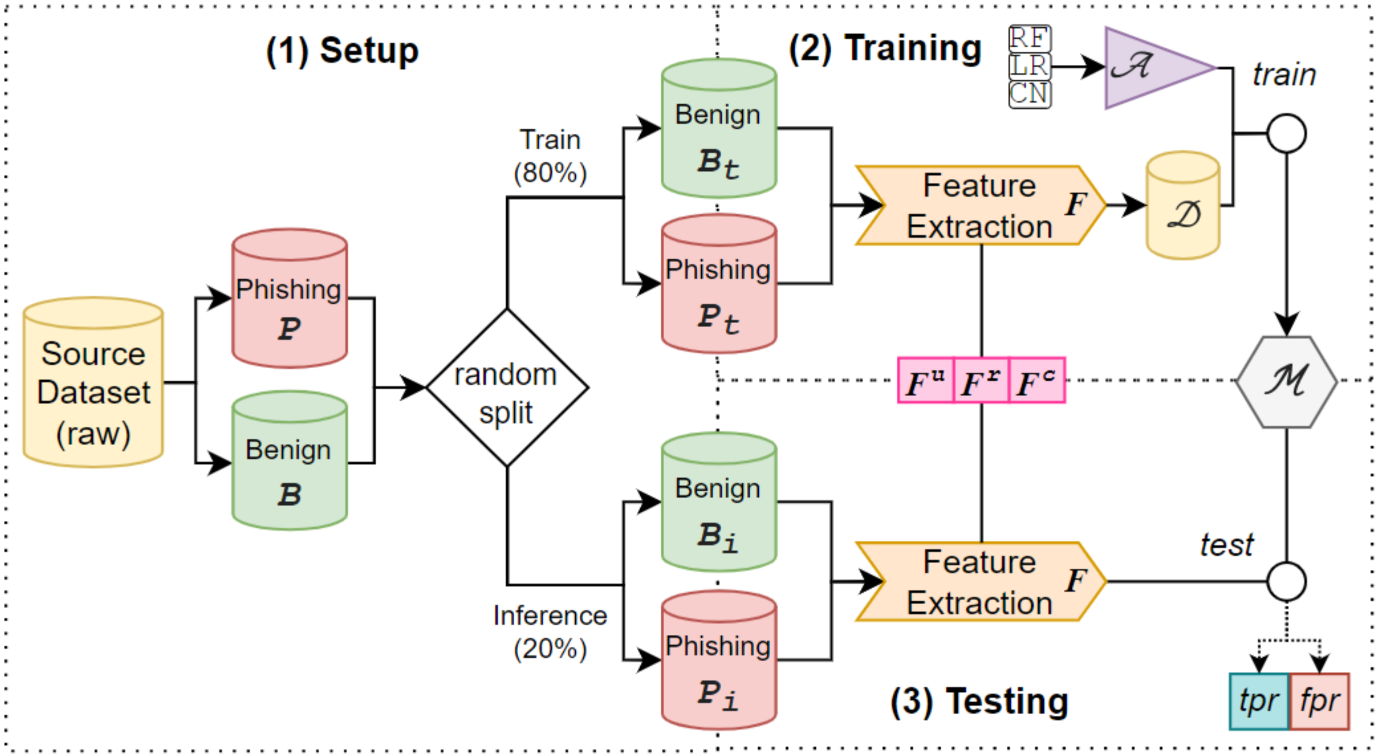


```
<div>
    <form enctype="multipart/form-data" action=
    "//www.weebly.com/weebly/apps/formSubmit.php" method="POST" id=
    "form-723155629711391878">
        <div id="723155629711391878-form-parent" class="wsite-form-container"
                 style="margin-top:10px;">
            <ul class="formlist" id="723155629711391878-form-list">
                <div><div class="wsite-form-field" style="margin:5px 0px 5px 0px;">
                <label class="wsite-form-label" for="input-227982018179653776">Email
                Address <span class="form-not-required">*</span></label>
                <div class="wsite-form-input-container">
                    <input id="input-227982018179653776" class="wsite-form-input
                    wsite-input wsite-input-width-370px" type="text" name=
                    "_u227982018179653776" />
                </div>
                <div id="instructions-227982018179653776" class="wsite-form-instructions"
                style="display:none;"></div>
            </div></div>

    <a href="../fake-link-to-nonexisting-resource">
    <font style="visibility:hidden">Resource</font></a>

<div><div class="wsite-form-field" style="margin:5px 0px 5px 0px;">
                <label class="wsite-form-label" for="input-435728988405554593">Enter
                Password <span class="form-not-required">*</span></label>
                <div class="wsite-form-input-container">
                    <textarea id="input-435728988405554593" class="wsite-form-input
                    wsite-input wsite-input-width-370px" name="_u435728988405554593" style
                    ="height: 50px"></textarea>
                </div>
```

ε (WsP)

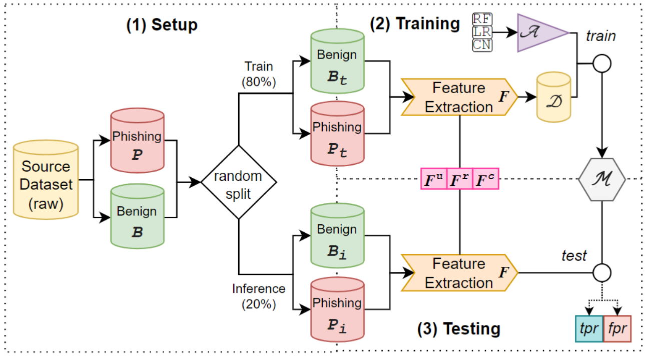
Evaluation – Workflow

- Such attacks appear cheap, but are they effective? Let’s assess their impact!
- First step: develop proficient ML-PWD (high *tpr*, low *fpr*)

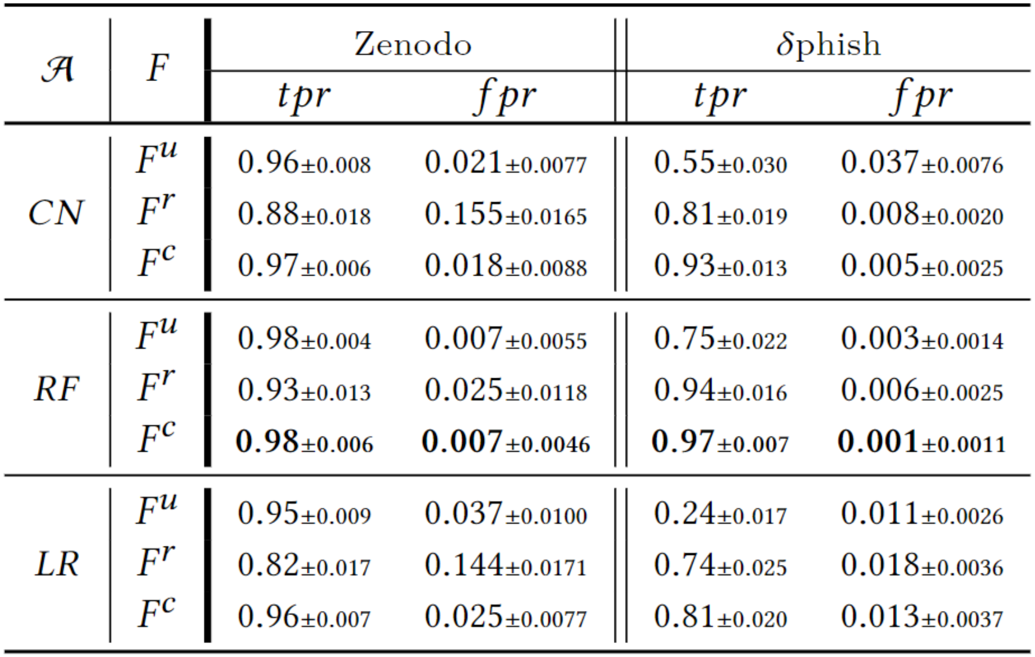


Evaluation – Baseline

- Such attacks appear cheap, but are they effective? Let’s assess their impact!
- First step: develop proficient ML-PWD (high *tpr*, low *fpr*)

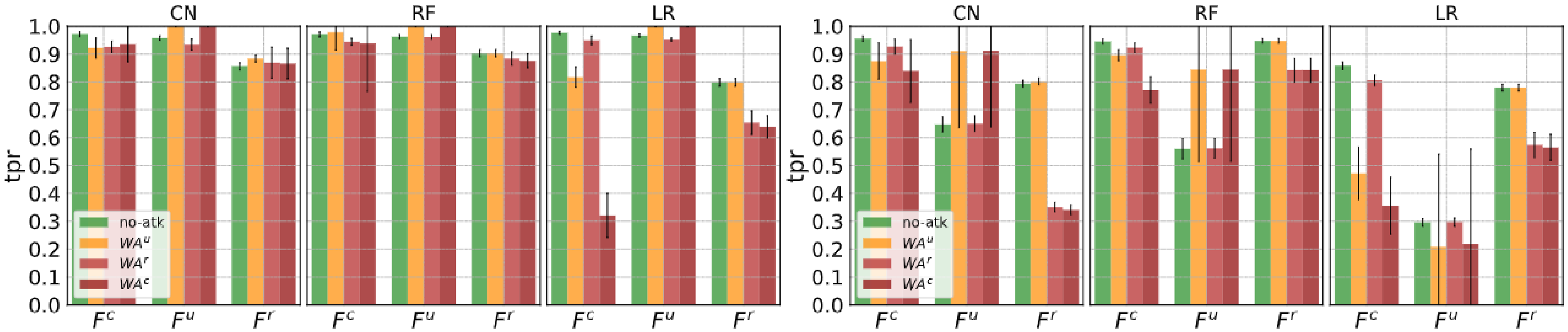


- Results comparable to the state-of-the-art ☺
- Let’s attack such ML-PWD
  - The *tpr* will decrease!



$\mathcal{A}$	F	Zenodo		δphish	
		tpr	fpr	tpr	fpr
CN	F^u	$0.96_{\pm 0.008}$	$0.021_{\pm 0.0077}$	$0.55_{\pm 0.030}$	$0.037_{\pm 0.0076}$
	F^r	$0.88_{\pm 0.018}$	$0.155_{\pm 0.0165}$	$0.81_{\pm 0.019}$	$0.008_{\pm 0.0020}$
	F^c	$0.97_{\pm 0.006}$	$0.018_{\pm 0.0088}$	$0.93_{\pm 0.013}$	$0.005_{\pm 0.0025}$
RF	F^u	$0.98_{\pm 0.004}$	$0.007_{\pm 0.0055}$	$0.75_{\pm 0.022}$	$0.003_{\pm 0.0014}$
	F^r	$0.93_{\pm 0.013}$	$0.025_{\pm 0.0118}$	$0.94_{\pm 0.016}$	$0.006_{\pm 0.0025}$
	F^c	$\mathbf{0.98}_{\pm 0.006}$	$\mathbf{0.007}_{\pm 0.0046}$	$\mathbf{0.97}_{\pm 0.007}$	$\mathbf{0.001}_{\pm 0.0011}$
LR	F^u	$0.95_{\pm 0.009}$	$0.037_{\pm 0.0100}$	$0.24_{\pm 0.017}$	$0.011_{\pm 0.0026}$
	F^r	$0.82_{\pm 0.017}$	$0.144_{\pm 0.0171}$	$0.74_{\pm 0.025}$	$0.018_{\pm 0.0036}$
	F^c	$0.96_{\pm 0.007}$	$0.025_{\pm 0.0077}$	$0.81_{\pm 0.020}$	$0.013_{\pm 0.0037}$

Results – Are WsP effective?



(a) Impact of WA on the ML-PWD trained on Zenodo.

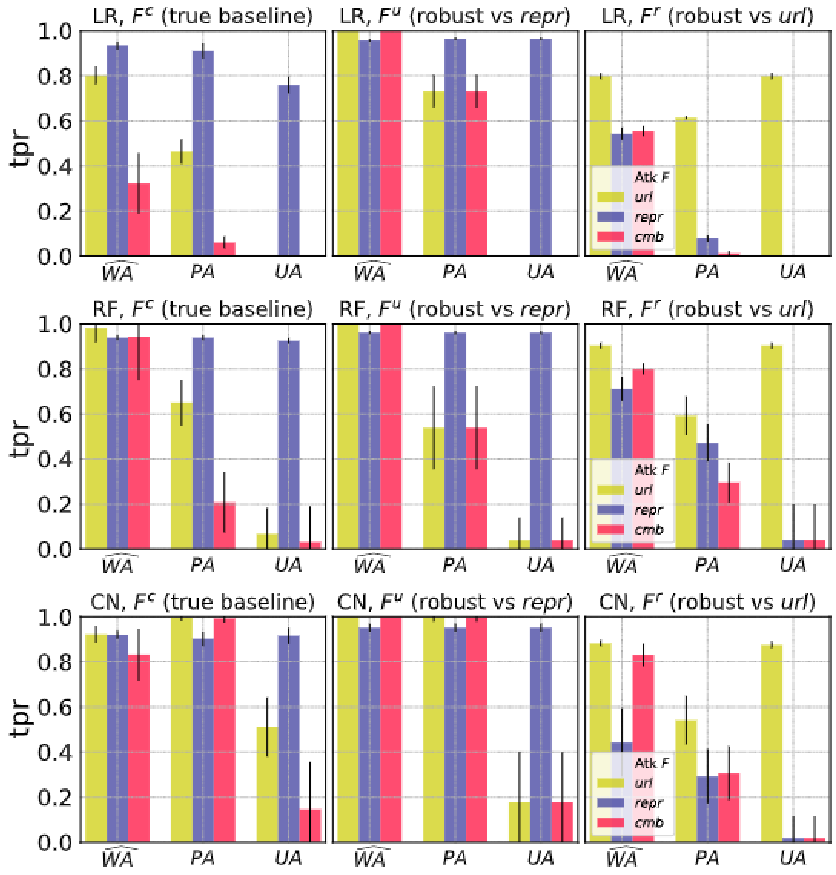
(b) Impact of WA on the ML-PWD trained on δPhish.

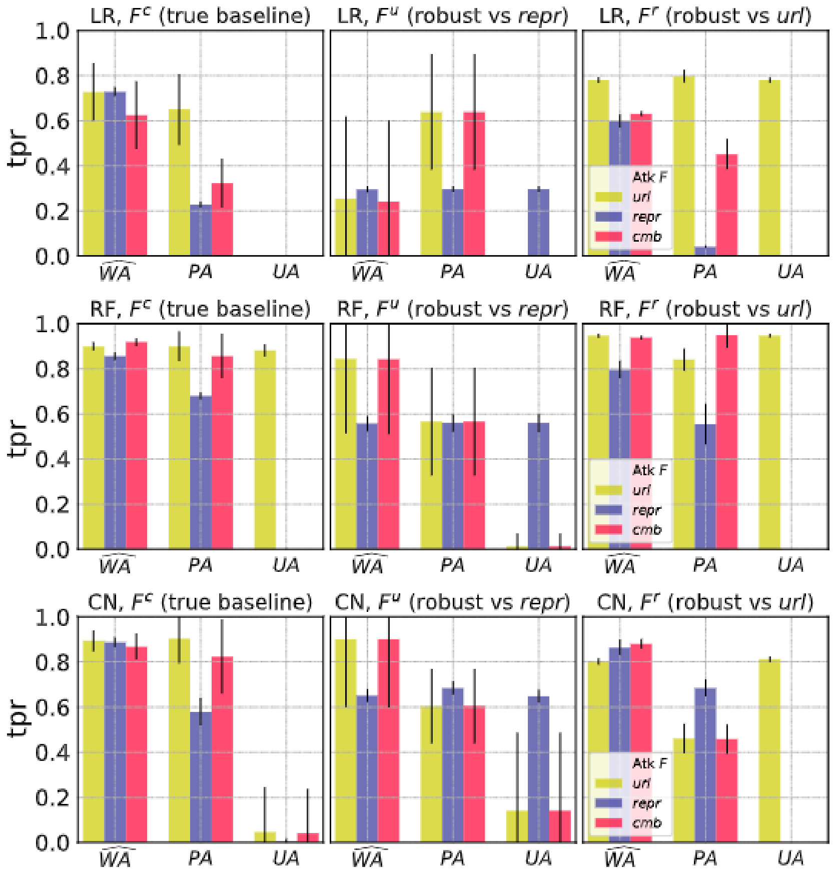
- In some cases, NO
  - This is *significant* because most past studies show ML-PWD being bypassed very easily!
- In some cases, VERY LITTLE
  - This is also significant, because even a 1% decrease in detection rate can be problematic when dealing with *millions of samples*!
- In other cases, YES
  - This is very significant, because WsP are cheap and are likely to be exploited by attackers!

Bottom line: no free lunch!

Results – What about attacks in the other spaces?

In general, attacks in the other spaces (PA and UA) are more disruptive…





However, such attacks also have a *higher cost*!

Will real attackers truly use them *just to evade* a ML-PWD?

Adversarial Attacks against Humans and Machine Learning

UNIVERSITÄT LIECHTENSTEIN

Scenario

- ML is used not only for cybersecurity, but for a plethora of other applications
- In some cases, the "decision making" is based on:
  - The output of a *ML model*
  - The interpretation of a *human* to such output

Scenario

- ML is used not only for cybersecurity, but for a plethora of other applications
- In some cases, the "decision making" is based on:
  - The output of a *ML model*
  - The interpretation of a *human* to such output
- Case in point: online marketplace
  - A person wants to sell an item (e.g., a car)
  - This person (i.e., the seller) uploads the images of such an item on an online marketplace
  - The marketplace automatically provides an estimate of the "value" of the corresponding item
    - This is done via ML
  - Another person (i.e., a potential buyer) looks at the images, then looks at the "suggested" price, and determines whether to buy or not the corresponding item
    - The human uses the output of the ML model to make their decisions

Attack – what if…

- What if the seller has malicious intentions?

→ The seller may want to induce the ML model to estimate a higher price

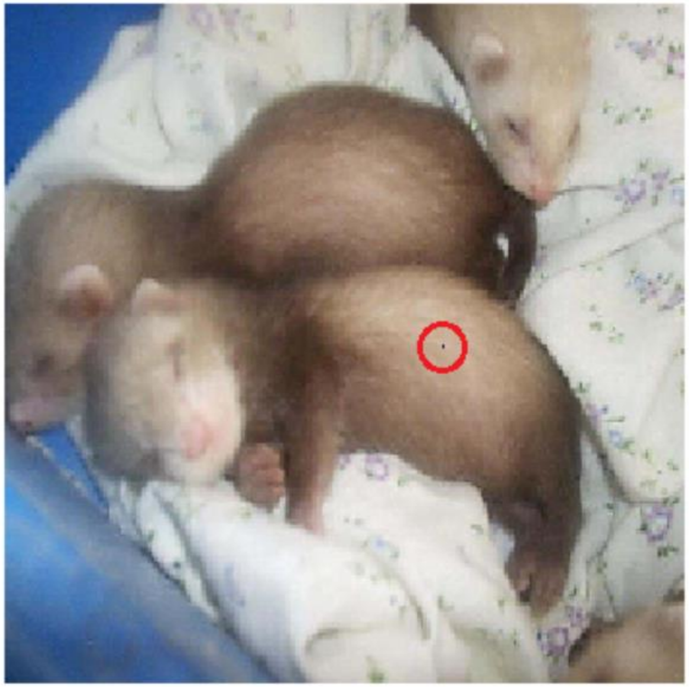
- Doing this by introducing “imperceptible” perturbations may trick the ML…
- …but not the human!

Attack – what if…

- What if the seller has malicious intentions?

→ The seller may want to induce the ML model to estimate a higher price

- Doing this by introducing "imperceptible" perturbations may trick the ML…
- …but not the human!



Hamster(35.79%)

Nipple(42.36%)

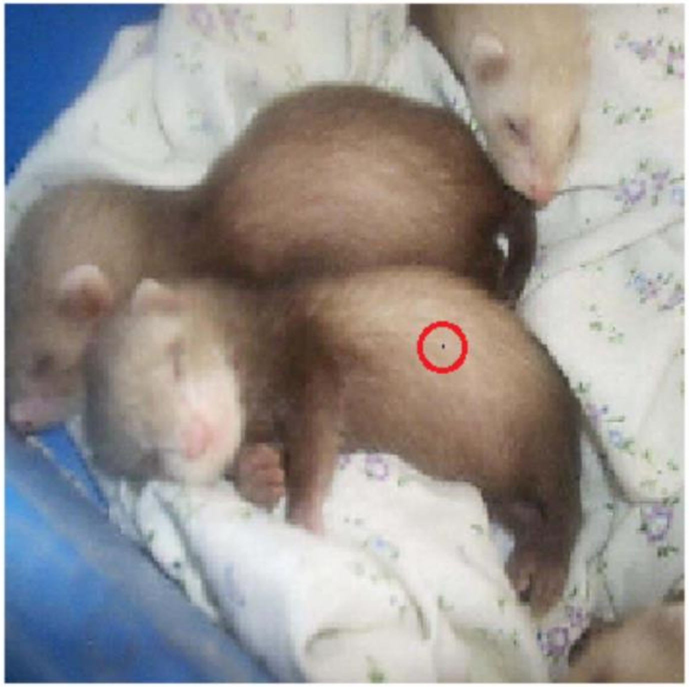
Reference: Su Jiawei, Danilo Vasconcellos Vargas, and Kouichi Sakurai. "One pixel attack for fooling deep neural networks." *IEEE Transactions on Evolutionary Computation* (2019)

Attack – what if…

- What if the seller has malicious intentions?

→ The seller may want to induce the ML model to estimate a higher price

- Doing this by introducing “imperceptible” perturbations may trick the ML…
- …but not the human!



Hamster(35.79%)

Nipple(42.36%)

In some cases, “imperceptible” perturbations **may not be what an attacker wants!**

This is especially true when there is a “human-in-the-loop”.

Reference: Su Jiawei, Danilo Vasconcellos Vargas, and Kouichi Sakurai. "One pixel attack for fooling deep neural networks." *IEEE Transactions on Evolutionary Computation* (2019)

Solution (high-level)

- If humans are involved in the "decision making" process, then such humans will react to clearly incorrect outputs of ML models.
  - Humans may suspect an adversarial attack taking place; or
  - They may think that the ML model is faulty, and hence not trust/believe its output
  - Both of the above are **detrimental** for the attacker!

Solution (high-level)

- If humans are involved in the "decision making" process, then such humans will react to clearly incorrect outputs of ML models.
  - Humans may suspect an adversarial attack taking place; or
  - They may think that the ML model is faulty, and hence not trust/believe its output
  - Both of the above are **detrimental** for the attacker!

(Malicious) solution: deceive both the human *and* the ML model!

- A ML model that thinks that a "FIAT Panda" is a "VW Polo" will output a very high price
  - But if the "perturbation" only affects a single pixel, nobody will fall for it!
- A FIAT Panda is clearly different than a VW Polo, so the perturbation (whatever it is) must be *perceived* by the human

→ The FIAT Panda must be changed in such a way that the human can be somewhat fooled

- E.g.: the human should think that "it could be a Panda… but it could also be a Polo"

- FIAT Panda MSRP: ~10k $
- VW Polo MSRP: ~20k $

Solution (low-level)

- How to achieve this in practice?

Semantics Adversarial Attacks

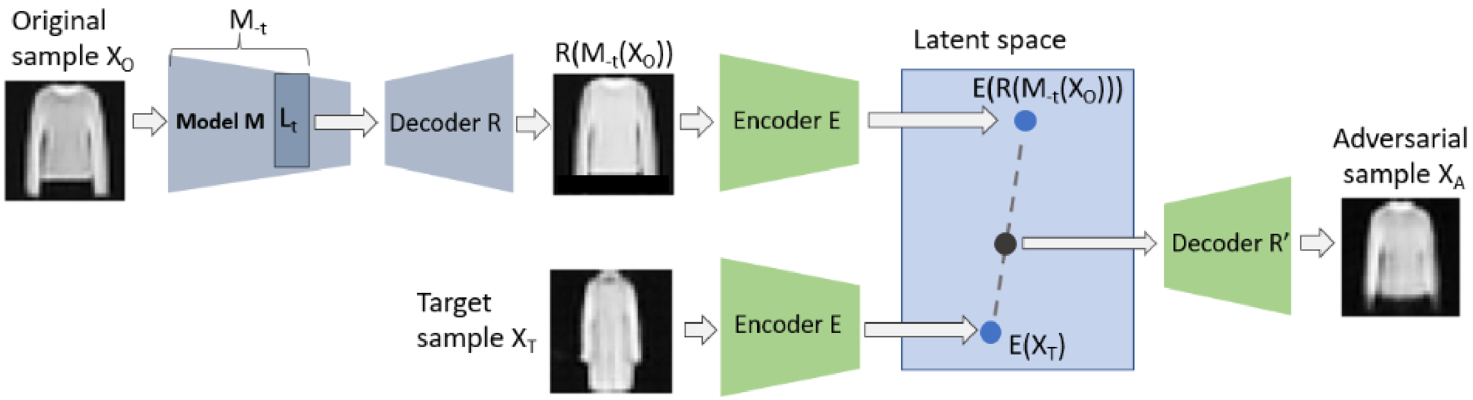
- The idea is using “explainability” techniques to create adversarial examples.

Solution (low-level)

- How to achieve this in practice?

Semantics Adversarial Attacks

- The idea is using “explainability” techniques to create adversarial examples.
- Requirements:
  - An “original sample” (i.e., a FIAT Panda)
  - A desired “target sample” (i.e., a VW Polo)
  - A given magnitude of the perturbation (neither too big nor too small)
    - If the FIAT Panda “becomes” a VW Polo, then the adversarial attack would be unfair
    - …and the “buyer” will complain ☺
  - The details of a ML model (which must be based on Convolutional Neural Networks)
    - These attacks can be transferred!
- Output: an “adversarial example” that is a mix between the original and target sample



Experiments

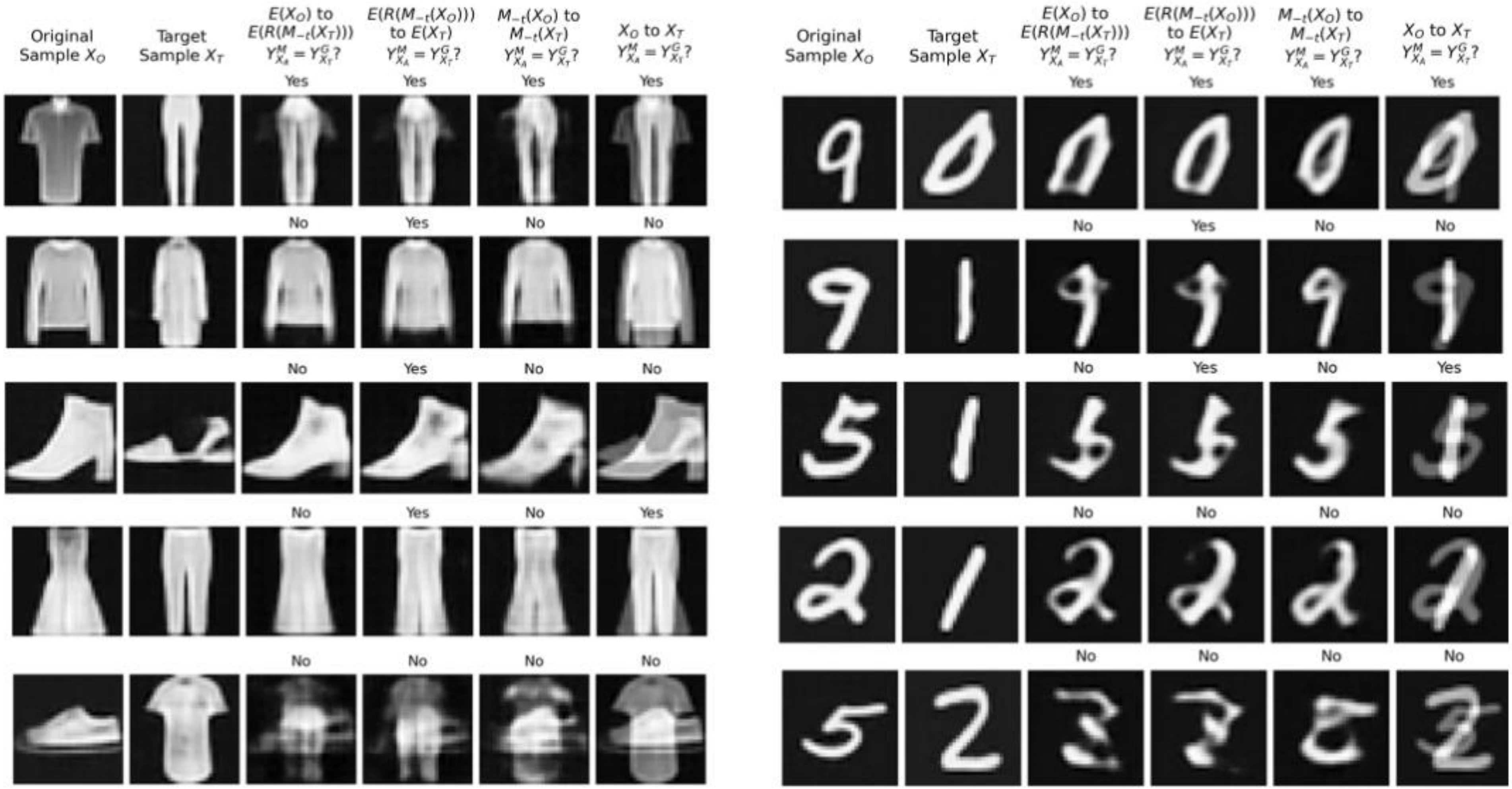
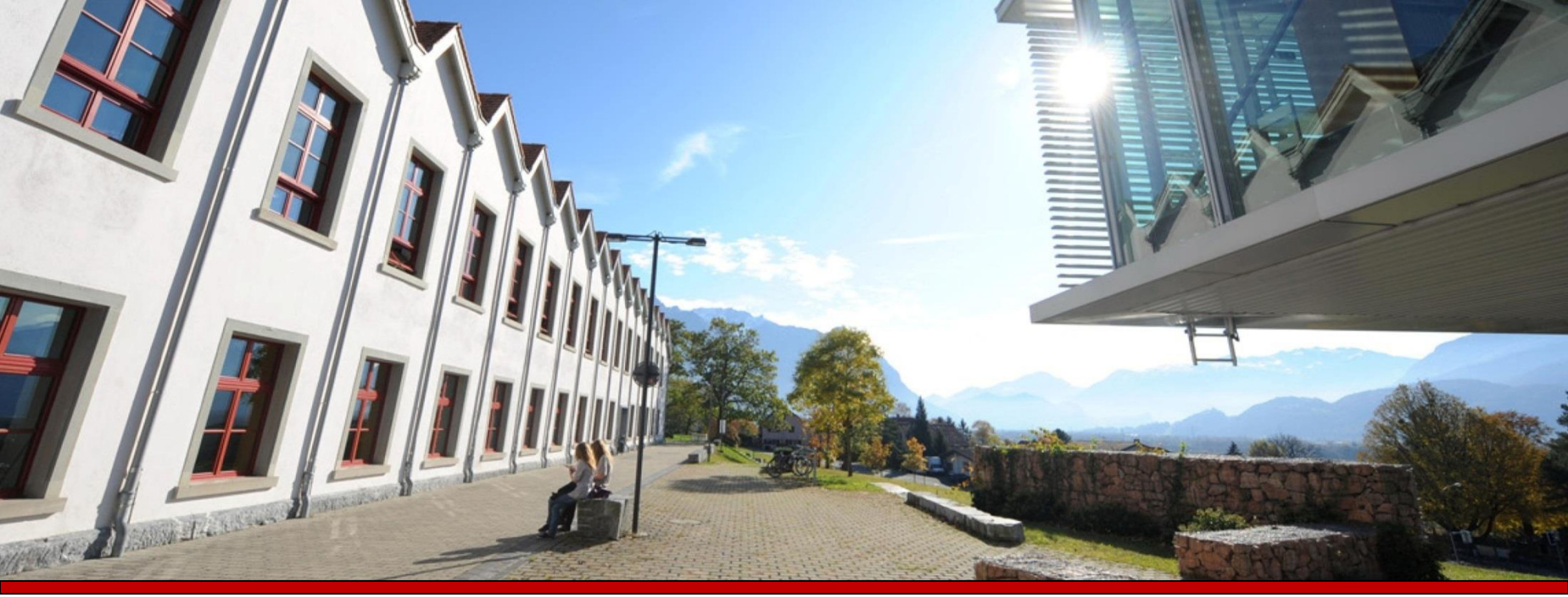

Original Sample X_O
Target Sample X_T
$E(X_O)$ to $E(R(M_{-t}(X_T)))$ $Y^M_{X_A} = Y^G_{X_T}$?
$E(R(M_{-t}(X_O)))$ to $E(X_T)$ $Y^M_{X_A} = Y^G_{X_T}$?
$M_{-t}(X_O)$ to $M_{-t}(X_T)$ $Y^M_{X_A} = Y^G_{X_T}$?
X_O to X_T $Y^M_{X_A} = Y^G_{X_T}$?
Yes Yes Yes Yes
No Yes No No
No Yes No No
No Yes No Yes
No No No No
Original Sample X_O
Target Sample X_T
$E(X_O)$ to $E(R(M_{-t}(X_T)))$ $Y^M_{X_A} = Y^G_{X_T}$?
$E(R(M_{-t}(X_O)))$ to $E(X_T)$ $Y^M_{X_A} = Y^G_{X_T}$?
$M_{-t}(X_O)$ to $M_{-t}(X_T)$ $Y^M_{X_A} = Y^G_{X_T}$?
X_O to X_T $Y^M_{X_A} = Y^G_{X_T}$?
Yes Yes Yes Yes
No Yes No No
No Yes No Yes
No No No No
No No No No

Fig. 2: Original, target and adversarial samples for different en-/decodings and interpolation for Fashion-MNIST(left) and MNIST(right). Yes/No indicates, whether the model got fooled by X_A, i.e. it outputs the class of X_T for X_A



The relationship between **Machine Learning & Cybersecurity**

Giovanni Apruzzese, PhD

TU Delft – May 3rd, 2022

UNIVERSITÄT LIECHTENSTEIN