Intrusion Detection through Unsupervised Machine Learning: pros, limitations, and workarounds

Andrea Bondavalli
University of Florence - Italy

RCL
RESILIENT COMPUTING LAB

In cooperation with: Andrea Ceccarelli, Tommaso Zoppi
Mohammad Gharib, Muhammed Atif,
Lorenzo Salani, Tommaso Capecchi

UNIVERSITÀ DEGLI STUDI FIRENZE
DIMAI
DIPARTIMENTO DI MATEMATICA E INFORMATICA "ULISSE DINI"

Outline

- Intrusion Detection. General intro and some background:
  - 0-day attacks, Anomaly versus Signature detection
  - Scoring Metrics, Attacks and Datasets.
  - An easy tool: RELOAD, Algorithms and comparison of their performance
- Observations and questions addressed here
- Feature Selection
- Meta-learning
- Performance with 0-days
- Conclusions

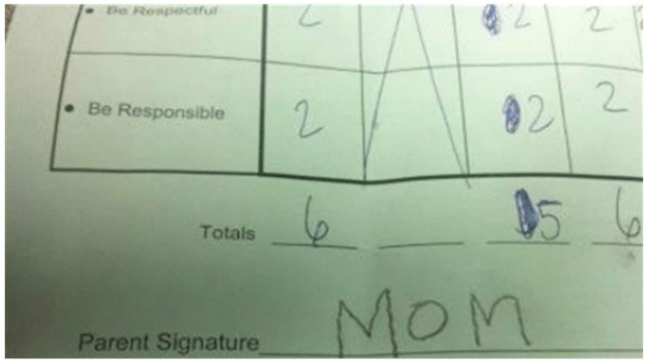
This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 823788.

Cyberattacks

Cyberattacks, with their ability to evolve, obfuscate and hide in between legitimate events, make them difficult to understand and analyse.

However they often leave some sign or distinguishing trace of their presence. A signature – or fingerprint – of each known attack can be derived and recorded.

"Not many clues, but we did find this fingerprint."



Different possibilities for ML algorithms (supervised).

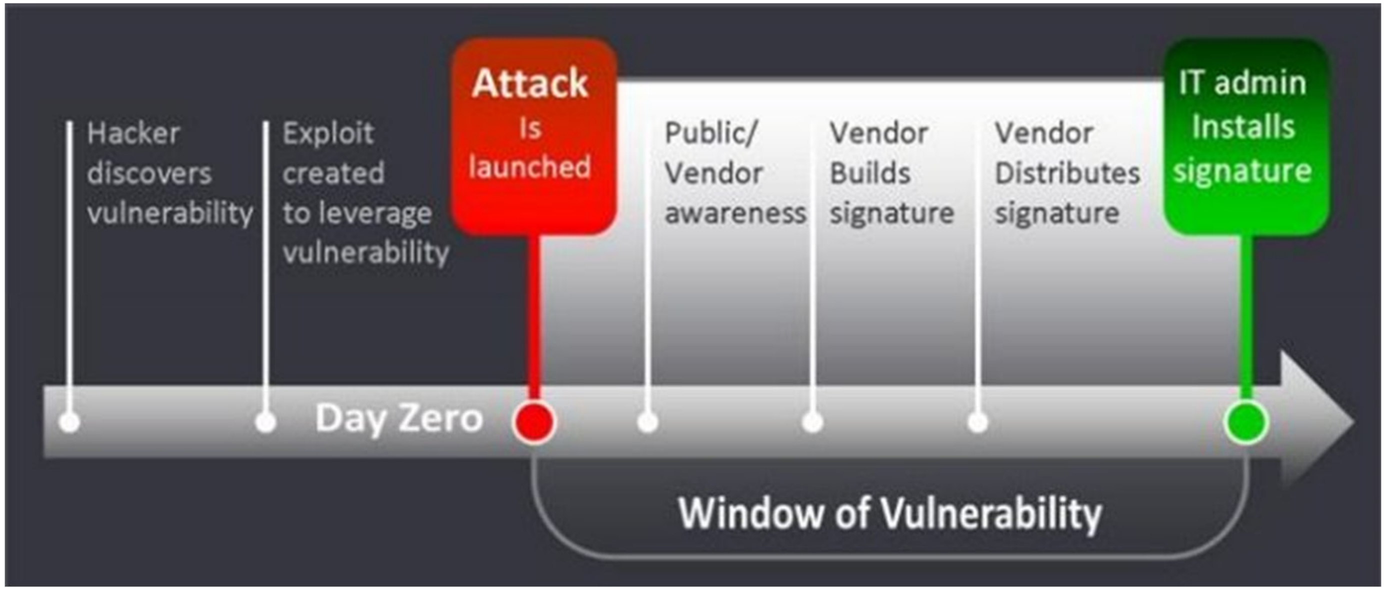
RESILIENT COMPUTING LAB

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 823788.

UNIVERSITÀ DEGLI STUDI FIRENZE
DIMAI DIPARTIMENTO DI MATEMATICA E INFORMATICA "ULISSE DINI"

Vulnerability Window

- Zero-day attack or vulnerability
- Signature-based algorithms cannot deal with them
  - Until the signature of the new attack is added
  - Could be too late: damaging actions already happened

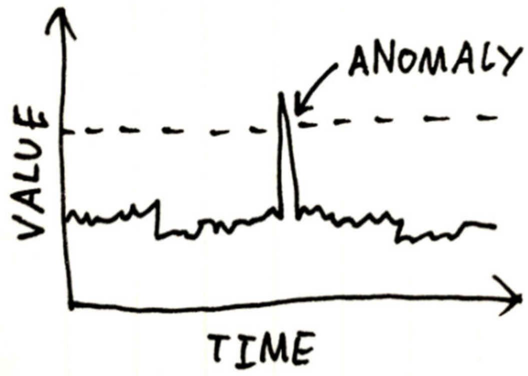


Anomalies

Corner cases??

What if something Unknown pops up?

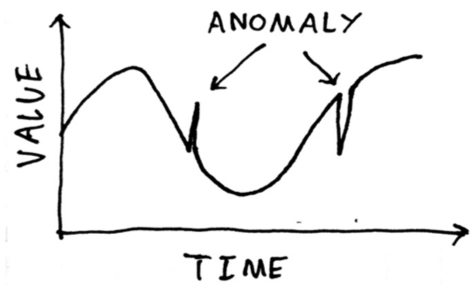
We still assume that an attack generates observable deviations from an expected – normal – behaviour.



This makes it possible to look and find patterns in data that do not conform to the expected behavior of a system: such patterns are known as Anomalies

Detecting such anomalies allows protecting against both known and zero-days attacks

(and corner cases not already encountered in safety critical systems)



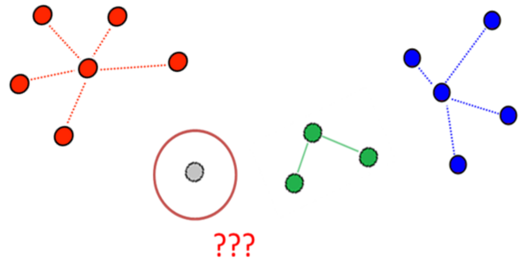
This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 823788.

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 823788.

Anomaly Detection

Anomalies in data translate to significant, and often critical, actionable information in a wide variety of application domains

- **Dependability**: Software errors, Misconfigurations
- **Security**: Malware, Attacks (e.g., DDoS/Ping Flood)
- **Safety**: unusual environment, corner cases, bad emergence in SoS



Anomaly detection refers to the problem of finding patterns in data that do not conform to an expected behaviour[1]

[1] Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." ACM computing surveys (CSUR) 41.3 (2009): 15.

Different possibilities for unsupervised ML algorithms.

Scoring Metrics (I)

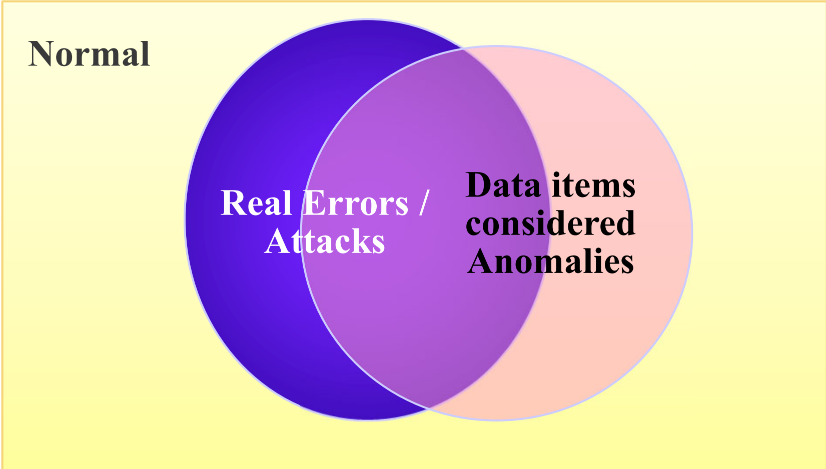
- The effectiveness of detection techniques are assessed depending on specific indicators.
- We start from basics:
- Given a data item and the judgement of an algorithm we may have one of 4 outcomes:

•**True Positive (TP):** erroneous behaviour recognized as such.

•**True Negative (TN):** real normal behaviour considered as such

•**False Positive (FP):** normal behaviour considered anomalous

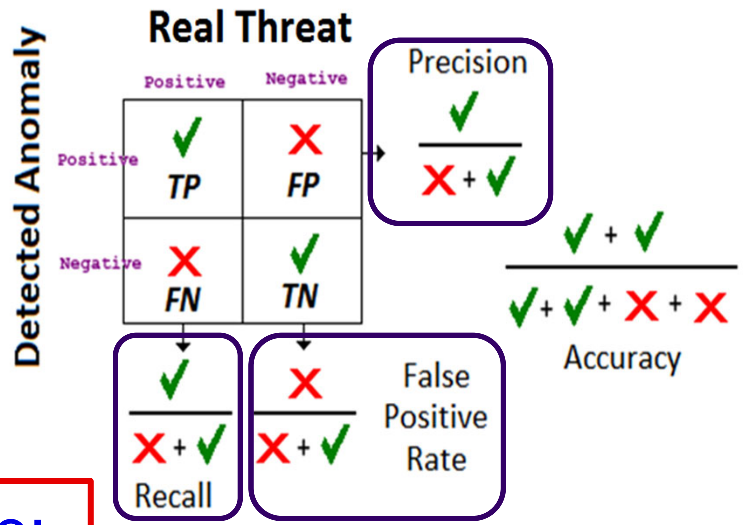
•**False Negative (FN):** erroneous behaviour considered normal



Scoring Metrics (II)

► Such individual items populate the **confusion matrix** on which metrics are derived

in our context
1) REAL NEGATIVES are much more than positives
2) Recall, Precision and their combination do not consider **TN** which is the most populated cell.



Other useful (less biased) metrics are:

$$FScore_{\beta}\,(F\beta) = (1+\beta^2)\frac{P \cdot R}{\beta^2 \cdot P + R}$$

$$F - Score(2)\; F2 = \frac{(1+4) * P * R}{4 * P + R}$$

$$\mathrm{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

RESILIENT COMPUTING LAB

Attacks and (Public) Datasets

- Usage of Public Data/Tools.
- Heterogeneous data sources, usual lack of documentation and the different strategies used to collect data may limit the understandability. Still public data and public tools allows reproducibility.
- Our baseline data:

Dataset			Attack Types	Dataset Features	
Index	Name	Year		Initial	Ordinal
D1	ADFANet	2015	5	11	3
D2	CICIDS17	2017	4	85	75
D3	CICIDS18	2018	5	85	75
D4	CIDDS	2015	4	16	7
D5	ISCX12	2012	4	16	6
D6	NGDIS-DS	2015	7	9	2
D7	NSLKDD	2009	4	42	37
D8	UGR16	2016	5	13	7
D9	UNSW-NB15	2015	8	45	38

Mapping of Attacks and Datasets

Attack Category	Malware	Web Attack	Web Application	Spam / Phishing	(D)Dos	BotNet	Data Breaches
ENISA Rank	1	2	3	4, 6	5	7	8
NSL-KDD	u2r		r2l		DoS		Probe
CTU-13						BotNet	
ISCX12		BruteForce			DoS, DDoS		Infiltration
UNSW-NB15	Worms	Fuzzers	Backdoor, Exploits, Shellcode		DoS		Analysis, Reconnaissance
UGR16				Blacklist, Spam	DoS	BotNet	Scan
NGIDS-DS	Malware, Worms		Backdoor, Exploits, Shellcode		DoS		Reconnaissance
Netflow-IDS				Mailbomb	Neptune, Portsweep		
AndMal17	Ransomware, Scareware			SMS, Adware			
CIDDS-001		BruteForce			DoS		PortScan, PingScan
CICIDS17		BruteForce			DoS (Slowloris, Goldeneye)		PortScan
CICIDS18		BruteForce (FTP, SSH)			DoS, DDoS	Bot	Infiltration

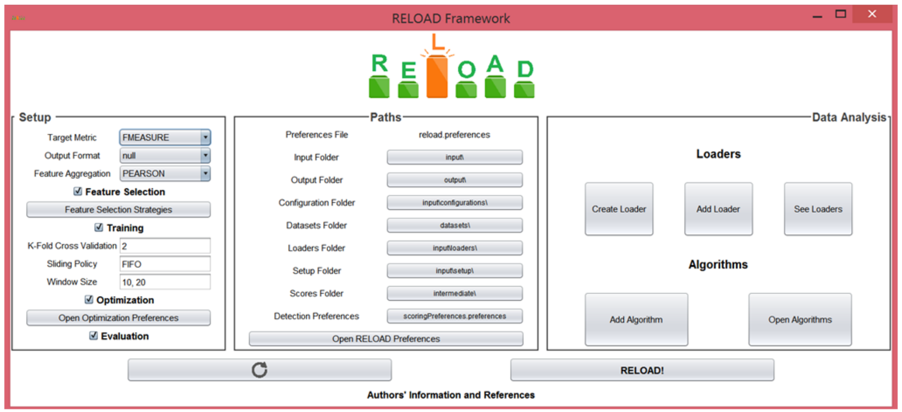
This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 823788.

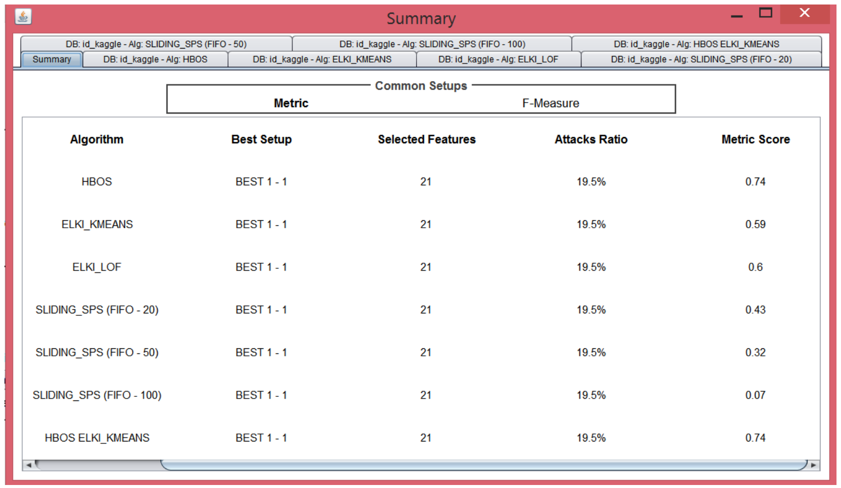
RELOAD: Rapid EvaLuation of Anomaly Detectors

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 823788.

A tool specifically crafted with attacks and errors datasets in mind to (among others)

- automatize analyses,
- help in devising the best parameter values
- allow fair comparisons,



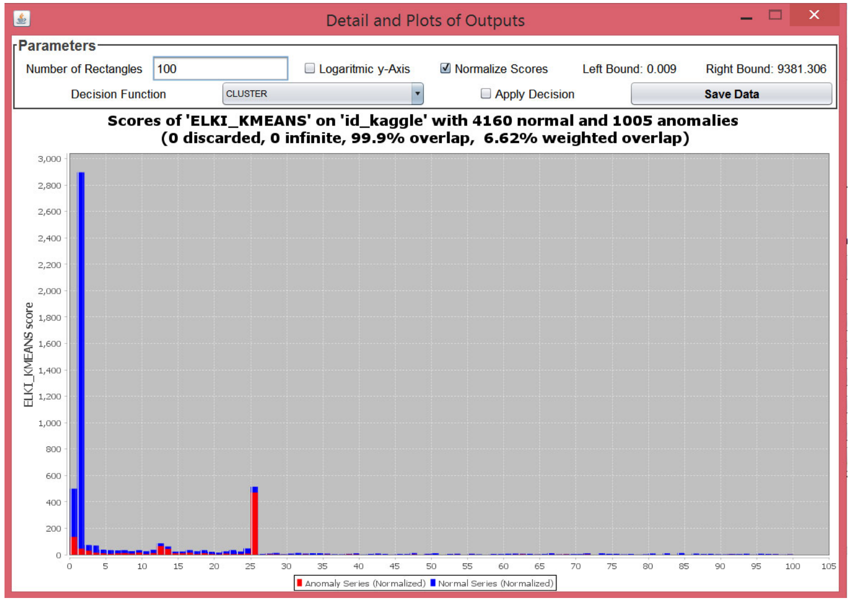


RCL
RESILIENT COMPUTING LAB

UNIVERSITÀ DEGLI STUDI FIRENZE
DIMAI DIPARTIMENTO DI MATEMATICA E INFORMATICA "ULISSE DINI"

Reload: Rapid EvaLuation of Anomaly Detectors (2)

- GUI; we tried to keep it as simple as possible
- Includes 10 features selection strategies, 17 algorithms, 11 metrics
- Includes the support for meta-learners



Zoppi, T., Ceccarelli, A., Bondavalli, A. Evaluation of Anomaly Detectors Made Easy with RELOAD. ISSRE 2019 (Tool paper)
Zoppi, T., Ceccarelli, A., Bondavalli, A. Into the unknown: Unsupervised machine learning algorithms for anomaly-based intrusion detection - Tutorial, DSN 2020

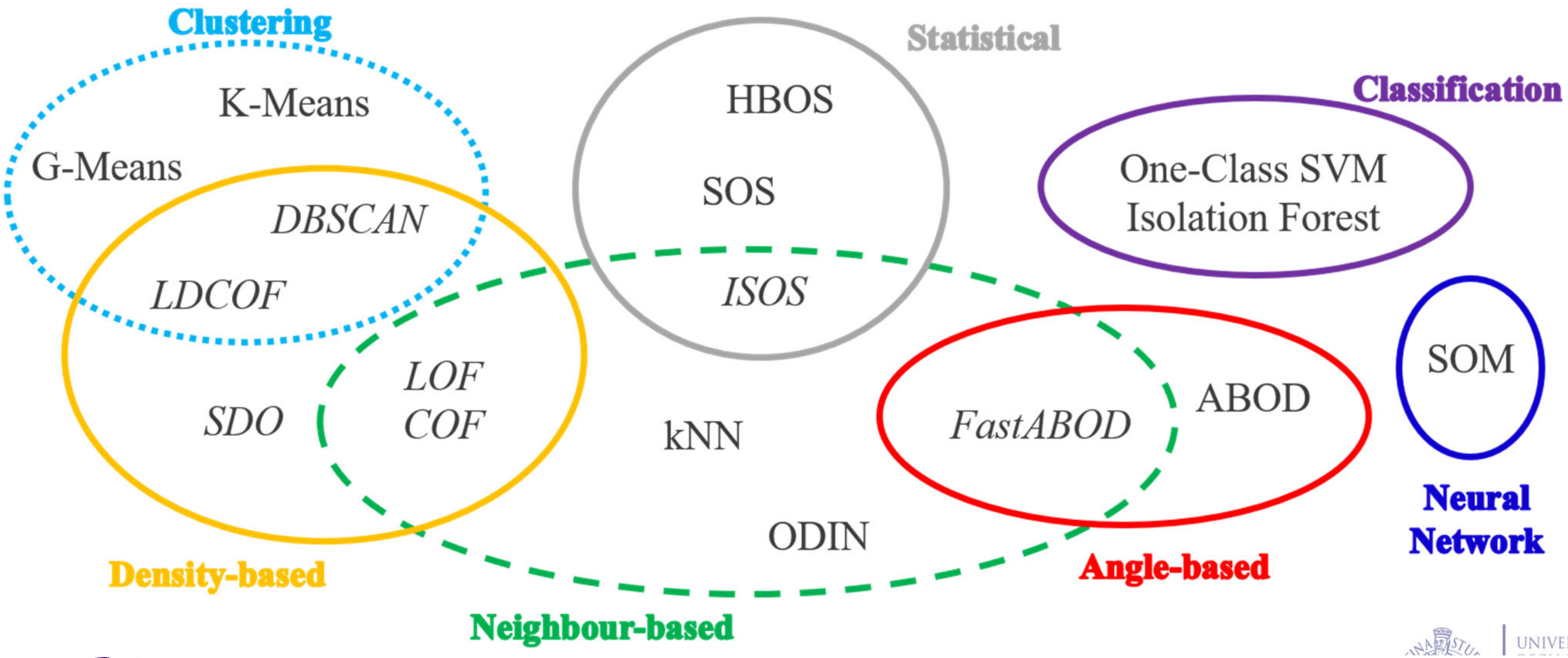
Downloadable at (GPL3 license): https://github.com/tommyippoz/RELOAD

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 823788.

Extensive comparison of unsupervised algorithms

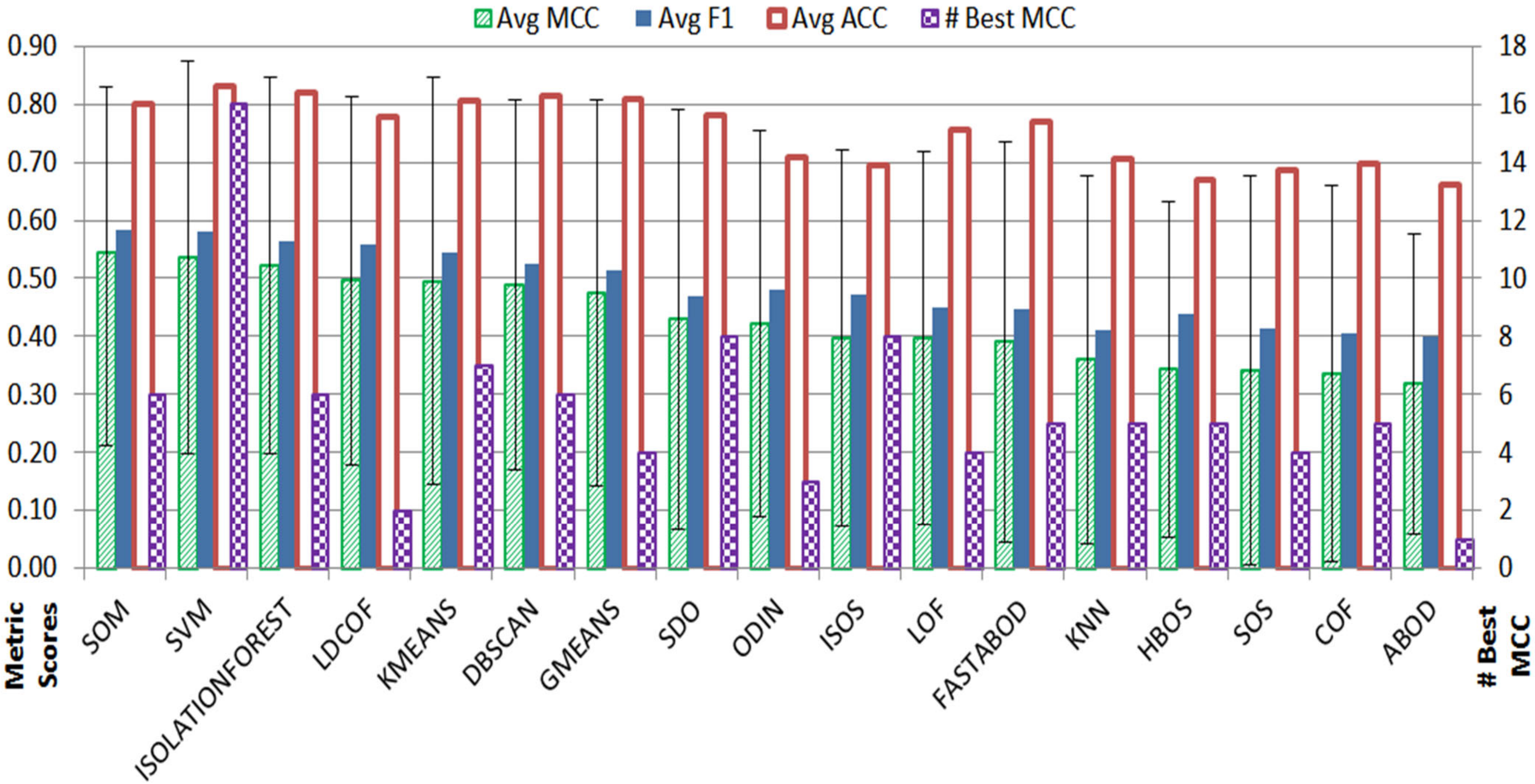
RELOAD exploited to investigate 17 algorithms belonging to the main families

using the attacks datasets



(a first version of this study appeared at ACM SAC 2019)

A sample of results



MCC as reference metric – good also for unbalanced datasets.

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 823788.

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 823788.

Attack driven Algorithm selection

How to select algorithm(s) that maximizes detection capability?

- We studied relations between attack families, anomaly classes and algorithms

Implications:

- an unknown attack belonging to an attack family is most likely to get observed by unsupervised algorithms that are particularly effective on such attack family.
- Consequently, rules can be defined to select algorithms based on "target" attack families

Fault/Attack Families → Anomalies of a certain type → Algorithms

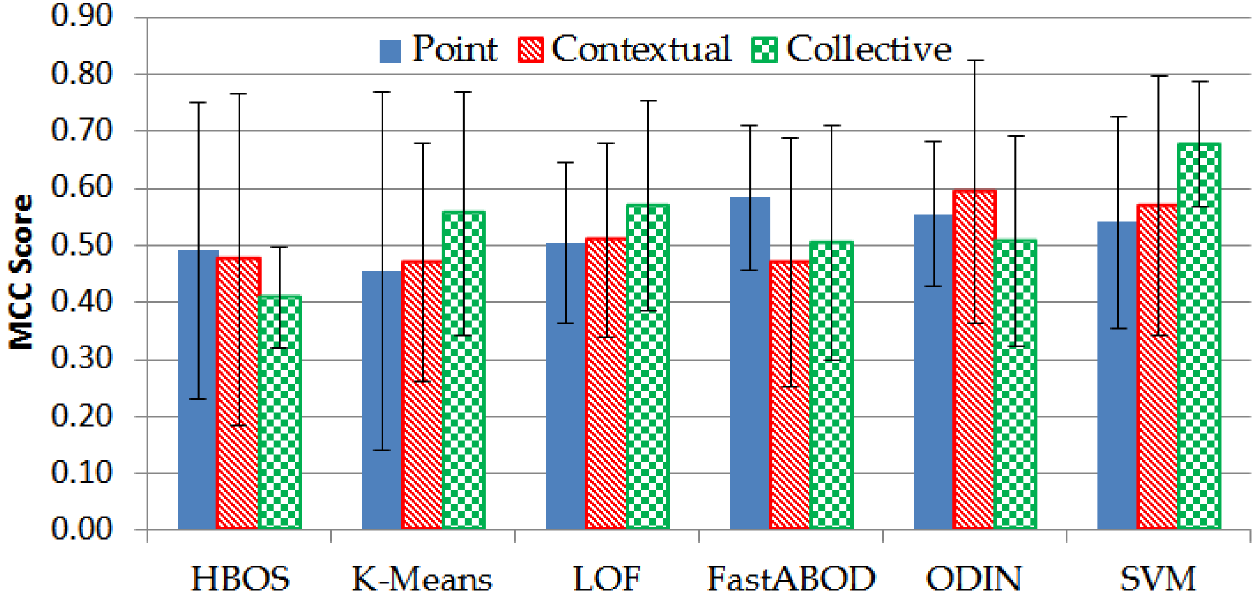
RESILIENT COMPUTING LAB

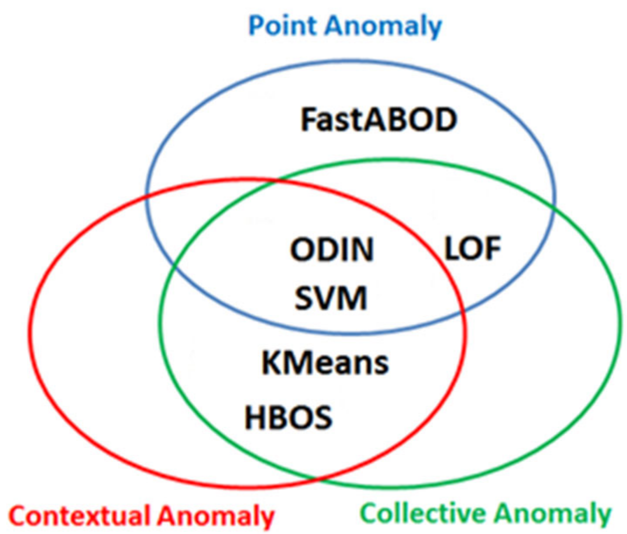
UNIVERSITÀ DEGLI STUDI FIRENZE
DIMAI DIPARTIMENTO DI MATEMATICA E INFORMATICA "ULISSE DINI"

Match algorithms to anomaly classes

We proceed in two steps:

- First, we run algorithms on synthetic datasets in which collective, contextual and point anomaly are introduced
- Then, we execute on real datasets





Zoppi, T., Ceccarelli, A., Salani, L., & Bondavalli, A. On the educated selection of unsupervised algorithms via attacks and anomaly classes. Journal of Information Security and Applications, Volume 52, 2020.

Observations

- Unsupervised algorithms are supposedly good to detect 0-days but are much weaker than their supervised counterpart for known threats…. Sometimes too weak to be useful at all

- Selecting and tuning an anomaly detector that minimizes misclassifications for a given problem/set-up is a substantial effort that requires to:
  - i) gather all the informative features i.e., system indicators and other measurable properties of the system,
  - ii) choose an unsupervised algorithm and,
  - iii) tune its hyper-parameters, to optimize its classification performance.

Questions explored here

- Q1: **Can I understand if the dataset at hand can be satisfacorily dealt with using unsupervised algorithms before doing all the work for selection and tuning??**

- Q2: **are there improvements on unsupervised learning able to improve detection capabilities and reduce the gap wrt supervised?**

- Q3: **is the unsupervised approach a proper and good response for 0-days?**

Q1: Selection of detectors

- Selecting and tuning an anomaly detector that minimizes misclassifications is far from easy.
- To improve detection performance literature recommends pre-processing **features** through ***filter-based* or *wrapper-based*** methods
- However, classification performance may still be not adequate **if the features do not contain enough information**.
-
- **In such cases the effort to identify the best anomaly detector will end up wasting time and money.**

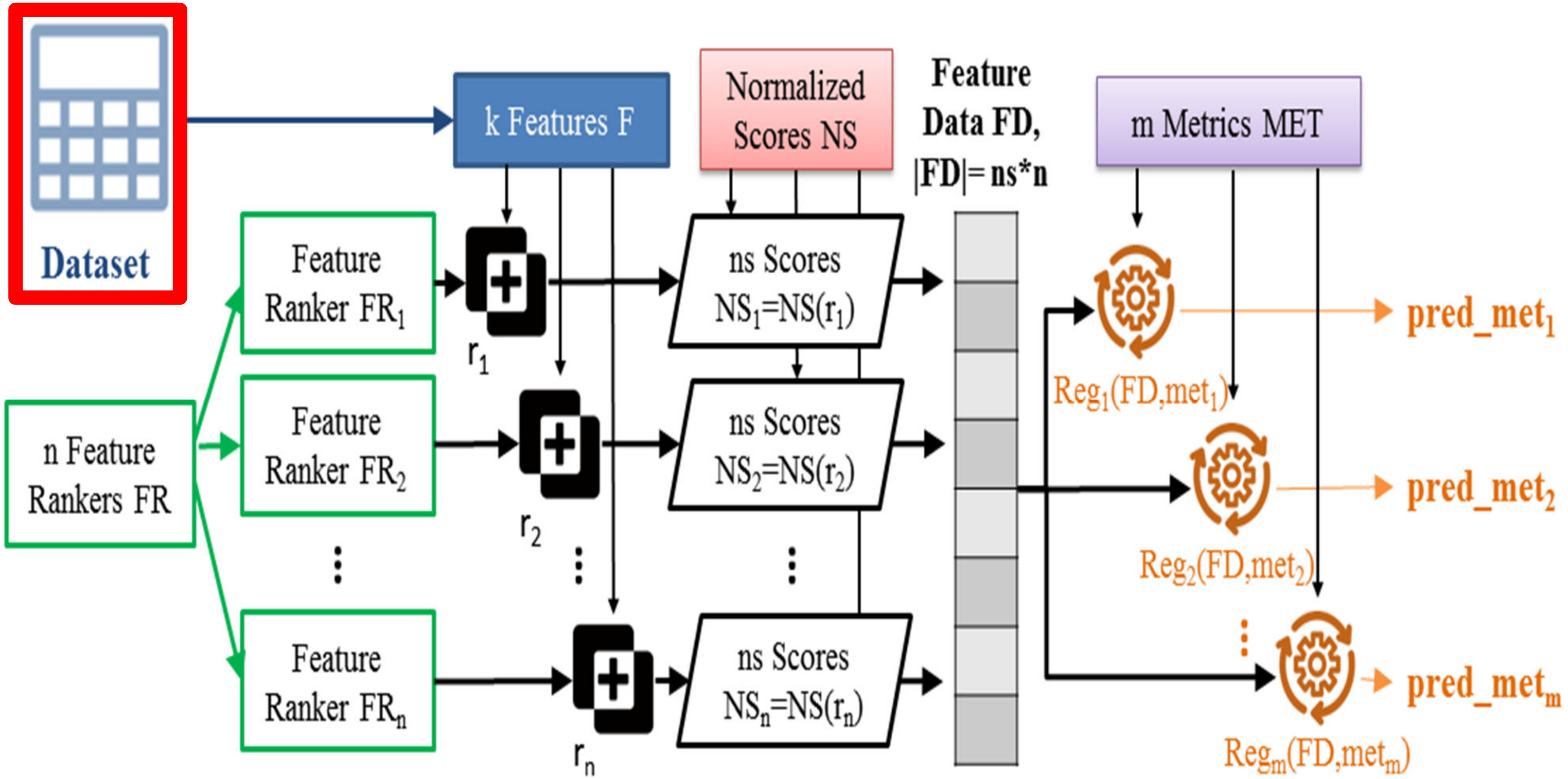
Approach to Q1: exploiting features

- We conjecture the **existence of a strong correlation** between
  - i) the scores that filter and wrapper-based feature rankers assign to features, and
  - ii) classification performance of anomaly-based intrusion detectors that use those features.
- Goal is to define a function that, using scores of feature rankers - before running any detector - predicts classification performance of unsupervised anomaly detection algorithms.
- If the features do not contain adequate information, misclassifications will be unacceptably high no matter the algorithm used.

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 823788.

Predicting Classificators Performance using Feature Ranking

- Our machinery to predict classification performance
  * Here Regressor are already trained



RESILIENT COMPUTING LAB

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 823788.

Feature Rankers

- We identify 8 feature rankers based on literature reviews.
  - FR1. Chi-Square;
  - FR2. ReliefF identifies differences of feature values between nearest neighbors;
  - FR3. Pearson Correlation between each feature and the label;
  - FR4. Information Gain measures the decrease in entropy when the feature is given with respect to when it is discarded;
  - FR5. PCA (Principal Component Analysis) analyzes the relationships among features and seeks the principal components through linear combination.
  - FR6 to FR8. 3 wrapper-based rankers based on Random Forests, J48, and OneR. They train tree-based classifiers, measure the impact that features have in building those trees or forests and use it to rank features.

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 823788.

Regressors

We chose **Supervised Regressors**

Supervised regressors build the RG set of ML algorithms intended to predict numeric values pred_met of a given metric met.

We adopted

- Linear Regression (LR) and Additive Regression (AR),
- Support Vector Machines (SVM) with Quadratic kernel,
- kNN-kStar,
- Random Forests (RF), and
- Multi-Layer Perceptron (MLP)

all implemented in WEKA.

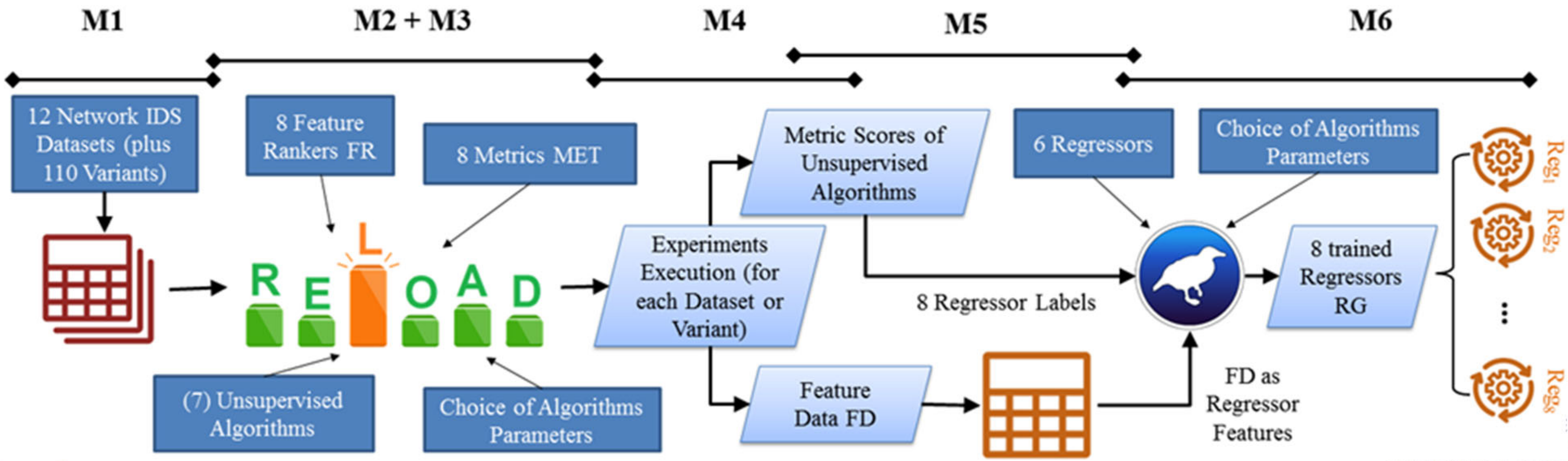
we choose regressors which rely on different mechanisms as neural networks, ensembles of decision trees, and other linear and non-linear ML algorithms.

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 823788.

To train the Regressors

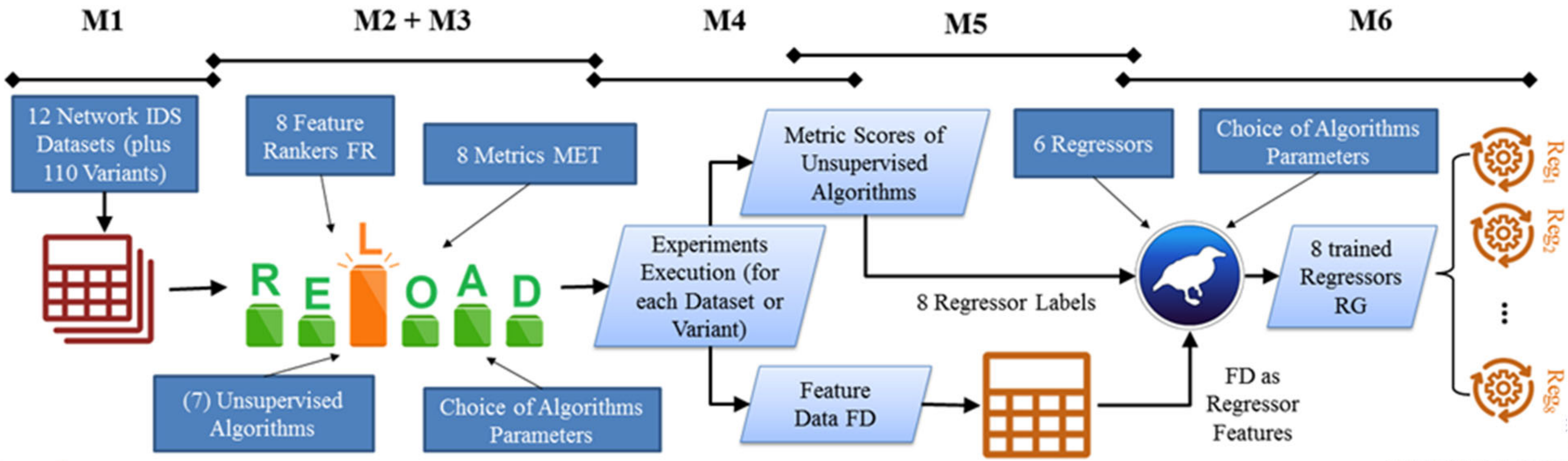
6 steps to a) verify correlation between feature rankers and classification metrics and b) build and train regressors RG to predict classification performance.

- M1. 12 public datasets elaborating on their features F. We also extract 110 variants.
- M2.8 commonly used filter and wrapper-based feature rankers Additionally, we define normalized scores NS.
- M3.unsupervised anomaly detection algorithms and metrics MET.



To train the Regressors-2

- M4.apply each unsupervised algorithm to each of the 122 datasets or variants, collecting scores of feature rankers and metric scores, and the R-Squared correlation between them.
- M5.Results of M4 build Feature Data FD and maximum metric values for each dataset, used as features and labels for the regressors.
- M6.train each regressor using FD as features and different metrics MET as labels. The best regressor for each metric met could then be used to calculate pred_met values



Correlation

- R-Squared correlations between normalized scores of individual (first 8 rows) or multiple (last 7 rows) **feature rankers** and metric scores (FPR, Precision, Recall, F1, F2, Accuracy, MCC and AUC) obtained by running the set of **unsupervised detectors**.

Normalized Score	Feature Ranker(s)	FPR	Precision	Recall	F1	F2	Accuracy	MCC	AUC
s_1	FR1 - Chi Squared	0.00	0.22	0.22	0.33	0.28	0.00	0.34	0.26
s_1	FR2 - ReliefF	0.01	0.07	0.04	0.07	0.04	0.06	0.09	0.09
s_1	FR3 - Pearson	0.01	0.20	0.30	0.36	0.36	0.04	0.31	0.27
s_1	FR4 - Info Gain	0.02	0.28	0.50	0.61	0.63	0.05	0.53	0.45
s_1	FR5 - PCA	0.01	0.00	0.02	0.00	0.01	0.05	0.00	0.00
s_1	FR6 - RandomForest	0.00	0.12	0.10	0.13	0.13	0.00	0.14	0.14
s_1	FR7 - OneR	0.09	0.00	0.02	0.00	0.02	0.68	0.01	0.00
s_1	FR8 - J48	0.02	0.17	0.32	0.37	0.39	0.10	0.30	0.24
s_1	All - FR	0.11	0.49	0.62	0.81	0.80	0.72	0.78	0.67
s_2	All - FR	0.11	0.47	0.59	0.75	0.75	0.58	0.71	0.60
s_3	All - FR	0.17	0.44	0.56	0.69	0.70	0.67	0.65	0.54
s_4	All - FR	0.17	0.43	0.55	0.68	0.69	0.69	0.63	0.54
$\{s_1, s_2\}$	All - FR	0.25	0.62	0.66	0.85	0.84	0.76	0.84	0.73
$\{s_1, s_2, s_3\}$	All - FR	0.35	0.64	0.72	0.87	0.85	0.83	0.86	0.77
$\{s_1, s_2, s_3, s_4\}$ = NS	All - FR	0.39	0.68	0.76	0.89	0.87	0.84	0.88	0.80

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 823788.

Regressors selection

The regressors were trained and then tested by submitting

i) FD as **features** and

ii) the value of one of the computed metrics FPR, P, R, F1, F2, ACC MCC and AUC as **label**.

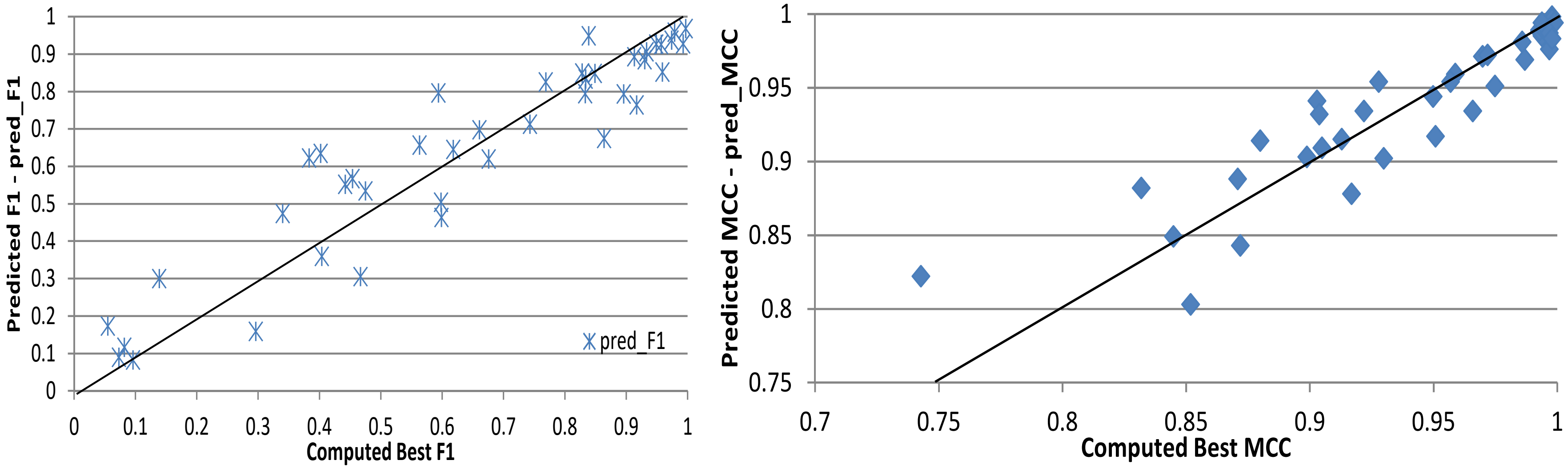
Average of relative residuals achieved for each metric.

Bold identifies the regressor that minimizes residuals.

Regressors were not able to predict FPR with satisfactory approximation (residual bigger than 1!) while for the other metrics residuals are very low and Random Forests is the one that achieves the lowest average!!!

	FPR	P	R	F1	F2	ACC	MCC	AUC
MLP	2.410	.206	.290	.221	.234	.035	.251	.145
AR	1.915	.153	.190	.194	.194	.031	.196	.087
RF	1.644	**.112**	**.155**	**.132**	**.115**	**.021**	**.138**	**.071**
LR	1.757	.155	.237	.210	.213	.027	.210	.095
kNN	1.870	.129	.164	.186	.155	.040	.216	.105
SVM	**1.532**	.160	.185	.179	.167	.024	.186	.086

Computed and predicted figures



- We used as Test Set 41 datasets or variants
- Computed and predicted F1 (left) and MCC (right) values for the Random Forest regressors.
- The black solid line graphically plots perfect correlation and helps showing residuals of each prediction.

Q2: Meta-learning

Often even the best unsupervised algorithm has too many misclassifications to satisfy the requirements of a critical system.

Several studies suggest that meta-learners may lower misclassifications.

However, this does not always result in improved capabilities: some misleading learners may drive meta-learners towards misclassifications.

⇩

Explore various **meta-learning approaches** with ensembles of unsupervised base-learners

to see if and how some specific meta-learning approach may significantly reduce misclassifications (with respect to non-meta unsupervised algorithms).

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 823788.

Categories of Metalearning

- **Single Classifier (SC)**: *Bagging, Boosting*. build ensembles of homogeneous base-learners, trained with different portions or feature sets extracted from the training dataset.
- **Multiple Classifiers (MC)**: *Stacking (Generalization), Voting (Weighted)*. heterogeneous base-level learners. aggregation of individual results does not depend on the order.
- **Multiple Classifiers with Ordering (MCO)**: *Cascading (Generalization), Delegating*. heterogeneous base-level classifiers. Final result based on subsequent operations therefore depends on the order.

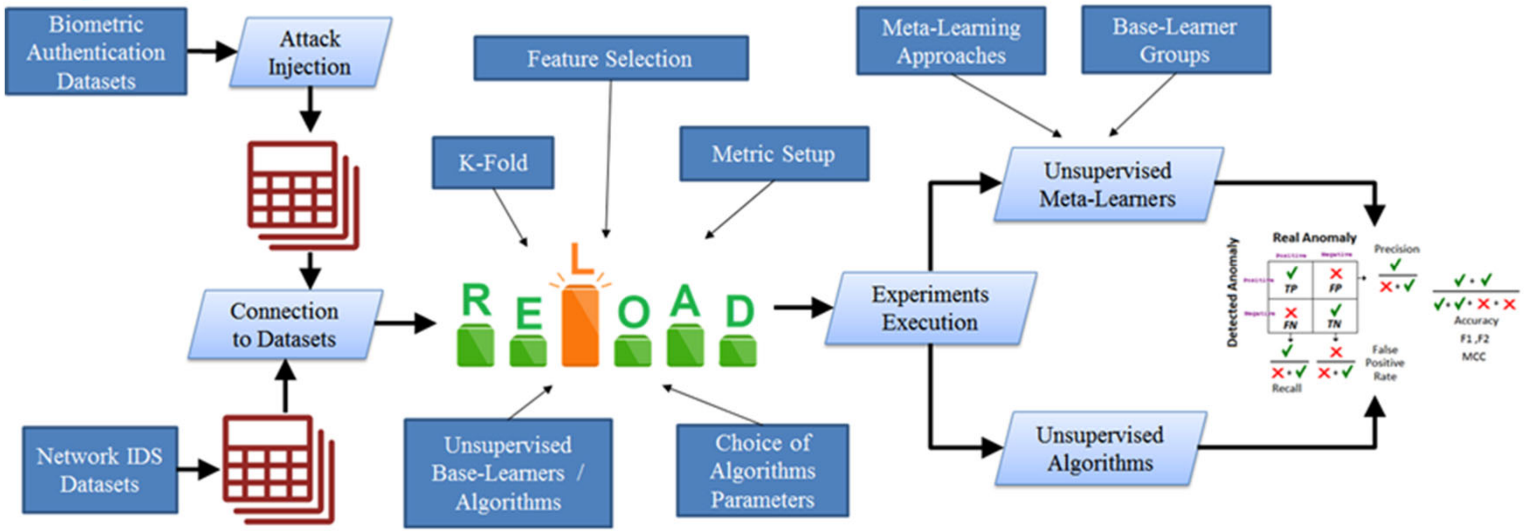
This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 823788.

Meta Learners

Meta-Learner	Category	(Meta)Features	Usage
Bagging	SC	Simple	Widespread
Boosting	SC	Simple	Widespread
Stacking	MC	Model-Based	Uncommon
Stacking Generalization	MC	Simple, Model-Based	Uncommon
Cascading	MCO		Uncommon
Cascade Generalization	MCO		Rare
Delegating	MCO	Simple	Rare
Voting	MC	Model-Based	Common
Weighted Voting	MC	Model-Based, Statistical	Uncommon

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 823788.

Our experiments



We performed a lot of experiments on our several dataset (including biometric datasets not reported here) and with many basic and meta-learners built upon the 17 unsupervised algorithms provided by RELOAD

We searched for optimal values of internal paramentes to maximise detection performace of each

Then evaluated base learners and meta-learners

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 823788.

Performance of the Meta-Learners

- Differences of MCC achieved by meta-learners on each dataset, wrt. the MCC achieved by best unsupervised (non-meta) algorithm.
  - Blank cells: meta-learner did not improve scores.
  - Bold underlined cells optimal classifier(s) for each dataset.

Dataset ID	Dataset	MCC Unsupervised BEST Algorithm	Bagging	Boosting	Voting	Weighted Voting	Stacking	Stacking Generalization	Delegating	Cascading	Cascade Generalization	#Meta Better Than Unsupervised
D1	ADFANet	0.98	0.002	**<u>0.006</u>**			0.004					3
D2	CICIDS17	0.91	**<u>0.02</u>**									1
D3	CICIDS18	0.90	0.08	**<u>0.10</u>**		0.08		0.07		0.09	0.09	6
D4	CIDDS	0.88		**<u>0.07</u>**			0.01	0.05				3
D5	ISCX12	0.51	0.01	**<u>0.18</u>**								2
D6	NGDIS-DS	0.39	0.19	**<u>0.44</u>**	0.15	0.24			0.12	0.13	0.15	7
D7	NSLKDD	0.79	0.002	**<u>0.06</u>**		0.003			0.002	0.01		5
D8	UGR16	0.31	**<u>0.28</u>**	**<u>0.28</u>**	0.27	0.27		**<u>0.28</u>**		0.21	0.21	7
D9	UNSW-NB15	0.57	**<u>0.08</u>**	0.04	0.02				0.01		**<u>0.08</u>**	5
Times Meta Better Than Unsupervised			**8**	**8**	**3**	**4**	**2**	**3**	**3**	**4**	**4**	
Times Meta Better Overall			**3**	**7**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	

RESILIENT COMPUTING LAB

UNIVERSITÀ DEGLI STUDI FIRENZE
DIMAI DIPARTIMENTO DI MATEMATICA E INFORMATICA "ULISSE DINI"

Remarks

Boosting outperforms base algorithms and other meta-learners in 7 out of the 9 datasets considered.

Adopting Boosting allows reaching the highest MCC scores, consequently minimizing misclassifications.

Other meta-learners (apart Bagging) are not even close to these numbers

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 823788.

Q3: detection performance in presence of 0-days

- We want to understand how well unsupervised learning (and meta learning) performs in scenarios where 0-days must be considered.
- For this we set up 2 specific experiments
  - i) to see how robust is unsupervised to 0-days and
  - ii) to compare with supervised

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 823788.

Robustness to 0-days

- F2-Score and MCC scores of SDO, HBOS, COF (boosting ensemble) and ODIN (bagging ensemble) on 8 different subsets of the SDN20 dataset. Each subset exposes different types of attacks and zero-days in the test set.

SDN20 subset	Train Set	Test Set		BASIC: SDO		BASIC HBOS		Boosting COF		Bagging ODIN	
	Attacks	Known Attacks	Zero-Days	F2 Score	MCC	F2 Score	MCC	F2 Score	MCC	F2 Score	MCC
SDN20_full	**DoS, DDoS, BFA, Probe, U2R**	**DoS, DDoS, BFA, Probe, U2R**	-	0.960	0.793	0.932	0.475	**0.995**	**0.986**	0.958	0.682
S1	**DoS, Probe, U2R**	**DoS, Probe, U2R**	**DDoS, BFA**	0.928	0.799	0.885	0.520	**0.995**	**0.986**	0.964	0.727
S2	**DoS, Probe, U2R**	-	**DDoS, BFA**	0.909	0.756	0.920	0.550	**0.990**	**0.983**	0.943	0.687
S3	**DoS, Probe, U2R**	-	**DDoS**	0.973	0.808	0.941	0.576	**0.998**	**0.992**	0.955	0.705
S4	**DDoS, BFA**	**DDoS, BFA**	**DoS, Probe, U2R**	0.956	0.792	0.782	0.532	**0.989**	**0.971**	0.958	0.682
S5	**DDoS, BFA**	-	**DoS, Probe, U2R**	0.917	0.733	0.731	0.472	**0.977**	**0.947**	0.913	0.699
S6	**DDoS, BFA**	-	**DoS, Probe**	0.918	0.737	0.733	0.472	**0.978**	**0.949**	0.915	0.697
S7	**DDoS, BFA**	-	**Probe**	0.918	0.734	0.736	0.477	**0.986**	**0.956**	0.918	0.699
			St.Dev	0.025	0.032	0.096	0.041	**0.008**	**0.018**	0.022	0.015

RCL
RESILIENT COMPUTING LAB

UNIVERSITÀ DEGLI STUDI FIRENZE
DIMAI DIPARTIMENTO DI MATEMATICA E INFORMATICA "ULISSE DINI"

How far are unsupervised?

- MCC score and Recall-Coverage restricted to 0-days for the best Supervised algorithm, the best unsupervised algorithm, and the best Meta-unsupervised algorithm (boosting).

Dataset	Attack Types				MCC			Recall-Unknowns		
	Train	Test	Unknown Attack Types	% 0-days in Test Set	Supervised	Unsupervised	Unsupervised Meta – Boost.	Supervised	Unsupervised	Unsupervised Meta – Boost.
CICIDS17	Patator, DoS, DDoS, PortScan	Patator, DoS, DDoS, PortScan	-	0.00	**0.9996**	0.9935	0.9959			
CICIDS17	DoS, DDoS, PortScan		Patator	0.98	**0.9818**	0.5744	0.9356	0.298	**0.995**	0.991
CICIDS17	Patator, DoS, DDoS		PortScan	11.17	0.8497	0.5958	**0.8634**	0.502	0.507	**0.626**
CICIDS17	Patator, DDoS, PortScan		DoS	17.72	**0.7137**	0.5539	0.5542	0.326	0.385	**0.566**
UGR	blacklists, nerisbotnet, anomaly-spam, dos, scan44	blacklists, nerisbotnet, anomaly-spam, dos, scan44	-	0.00	**0.9272**	0.8115	0.8718			
UGR	blacklists, anomaly-spam, dos, scan44		nerisbotnet	0.44	**0.9079**	0.8148	0.8684	0.000	0.000	**0.224**
UGR	blacklists, nerisbotnet, dos, scan44		anomaly-spam	0.71	**0.8947**	0.8090	0.8702	0.000	0.000	0.000
UGR	blacklists, nerisbotnet, anomaly-spam, scan44		dos	2.27	**0.8739**	0.8163	0.8326	0.505	0.501	**0.786**
UGR	blacklists, nerisbotnet, anomaly-spam, dos		scan44	9.17	0.5421	**0.7533**	0.6742	0.216	**0.999**	**0.999**

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 823788.

Conclusions

Intrusion Detectors (IDs) to deal with zero-day attacks.

Overviewed Unsupervised Machine Learning (ML) and tooling (**RELOAD**) for their assessment (including Datasets, Metrics, Parameters' Tuning)

- Predicting unsupervised anomaly detection algorithms performance using Feature Ranking - before running any detector !

- Improving unsupervised anomaly detection algorithms performance using meta-learning: in our experiments **Boosting** by far the best

- **0-days**: unsupervised very good in detection of 0-days, also very robust (very low standard deviation)

- Gap with supervised quite significant → we need to understand how to use together unsupervised (meta) and supervised algorithms.

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 823788.

Relevant papers

- Zoppi, T., Ceccarelli, A., Bondavalli, A. Evaluation of Anomaly Detection Algorithms Made Easy with RELOAD. International Symposium on Software Reliability Engineering, ISSRE, 2019, pp. 446–455,
- Zoppi, T., Ceccarelli, A., Bondavalli, A. Into the unknown: Unsupervised machine learning algorithms for anomaly-based intrusion detection Proceedings - 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks: Supplemental Volume, DSN-S 2020, 2020, pp. 81
- Zoppi, T., Ceccarelli, A., Salani, L., & Bondavalli, A. On the educated selection of unsupervised algorithms via attacks and anomaly classes. **Journal of Information Security and Applications**, Volume 52, 2020.
- T. Zoppi, A. Ceccarelli, T. Capecchi, A. Bondavalli. Unsupervised Anomaly Detectors to Detect Intrusions in the Current Threat Landscape. **ACM/IMS Trans. Data Sci**. 2, 2, Article 7 (April 2021), 26 pages.
- Zoppi, A. Ceccarelli and A. Bondavalli. MADneSs: A Multi-Layer Anomaly Detection Framework for Complex Dynamic Systems. **IEEE Transactions on Dependable and Secure Computing**, vol. 18, no. 2, pp. 796-809, 1 March-April 2021.
- T. Zoppi, M. Gharib, M. Atif, A. Bondavalli. Meta-Learning to Improve Unsupervised Intrusion Detection in Cyber-Physical Systems. **ACM Transactions on Cyber-Physical Systems**, in press.
- T. Zoppi, A. Ceccarelli, A. Bondavalli. Unsupervised Algorithms to Detect Zero-Day Attacks: Strategy and Application. **IEEE Access**, in press .
- T. Zoppi, A. Ceccarelli, A. Bondavalli. Feature Rankers to Predict Classification Performance of Unsupervised Intrusion Detectors. Submitted manuscript

RESILIENT COMPUTING LAB

UNIVERSITÀ DEGLI STUDI FIRENZE
DIMAI DIPARTIMENTO DI MATEMATICA E INFORMATICA "ULISSE DINI"

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 823788.