



# *Generating synthetic traffic to improve the robustness of network intrusion detection*

**Gregory Blanc**
*IMT/Télécom SudParis, Institut Polytechnique de Paris*

**LSE Summer Week 2022**
EPITA, Kremlin-Bicêtre, 2020/07/07

# $whoami

- faculty at **Télécom SudParis**, an **IMT** school, member of **IP Paris**
- researcher at **SCN** *(Sécurité et Confiance Numérique)*, a team of the **SAMOVAR** lab
- head of the **SSR** *(Sécurité des Systèmes et Réseaux)* specialization curriculum
- interested in **network security**, **network virtualization**, **machine learning for cybersecurity**
- holds a Ph.D degree from *Nara Institute of Science and Technology* (**NAIST**), Japan
- holds a *Mastère Spécialisé* in *Networks and Information Security* and a *Diplôme d'Ingénieur* from **ESIEA**
- led the **SWAN** *(Security of Web ApplicatioNs)* WG at **WIDE**, Japan
- worked as a security solutions integrator at **BT CyberNetworks**

# Team and Projects

## Contributors

- Mustafizur R. Shahid (Ph.D, 2017–2021)
- Cuong Pham (Engineer, 2017–2018)
- Houda Jmila (Postdoc, 2018–)
- Marwan Lazrag (Engineer, 2019–)
- Paul Peseux (Trainee, 2019)
- Paul-Henri Mignot (Engineer, 2021–)
- Adrien Schoen (Ph.D, 2021–)
- Solayman Ayoubi (Ph.D, 2022–)

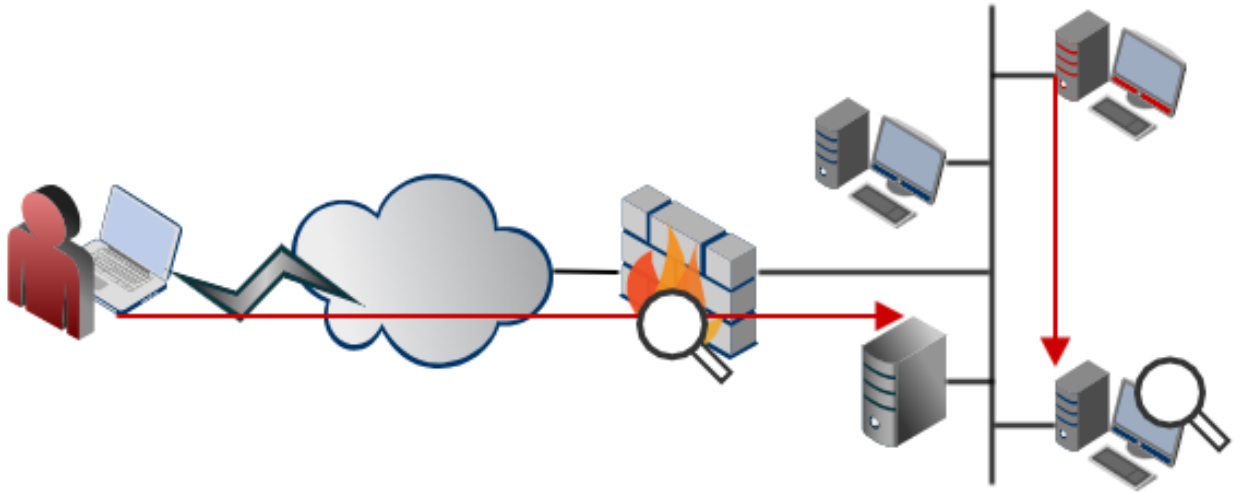
## Projects

- *CEF VARIoT* (Vulnerability and Attack Repository for IoT, 2019–2022)
- *H2020 SPARTA – CAPE* (Continous Assessment in Polymorphous Environments, 2019–2022)
- *ANR GRIFIN* (Cognitive and Programmable Security for Resilient Next-Generation Networks, 2021-2025)
- *Futur & Ruptures Ph.D Grant* (IMT, 2017–2021)

Collaborators: Hervé Debar, Pierre-François Gimenez (IRISA), Yufei Han (Inria), Christophe Kiennert, Frédéric Majorczyk (DGA-MI), Ludovic Mé (Inria), Thomas Silverston (LORIA), Sébastien Tixeuil (LIP6), Zonghua Zhang

# Outline

# Intrusion Detection



**Alert** on any **suspicious** activity enabling later filtering or correlation

- What is suspicious?
  - **misuse**: *activity known to be malicious*
  - **anomaly**: *activity deviant from normal*
- How to capture suspicious activities?
  - at the host: process, log, file, etc.
  - in the network: flow, packet headers, contents, etc.

*Huge* volume of activities incur *longer* processing time

# Misuse detection

| | |
|---|---|
| Approach | *mostly* attack signatures |
| Features | packet headers, flow stats, TCP connections, etc. |
| Trends | data mining and machine learning on labeled traffic datasets |
| Challenges | ■ lack of datasets (existence, diversity, freshness, reliability)<br>■ frequency of model re-training |

# Anomaly detection

Approach (normal) behavioural profiles

Learning unsupervised, semi-supervised, supervised

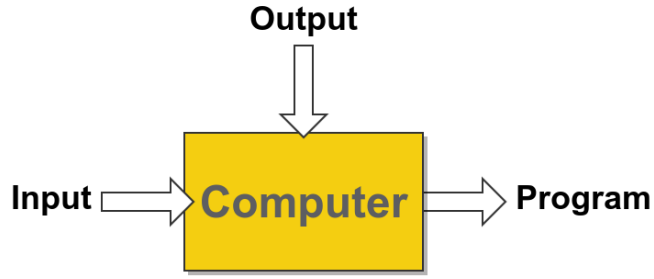
Challenges
- cleanliness of datasets
- accuracy of normal behaviour
- high false positive rate


Works well with *low-entropy* normal behaviour

# Outline

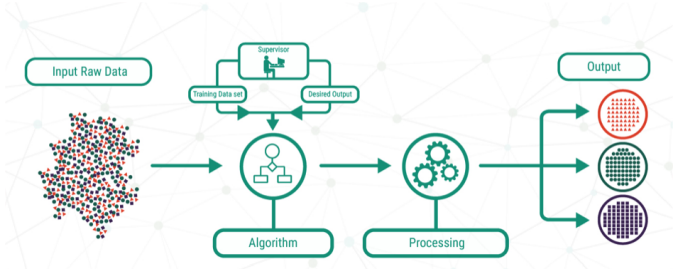
# Machine Learning


(a) Experience, task and performance.
Source: `underscore.vc`

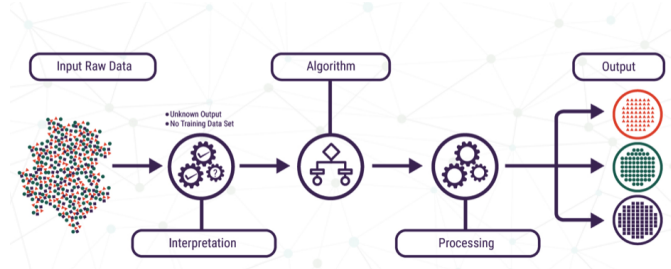



(b) Input, output, program

Figure: Two definitions of machine learning.



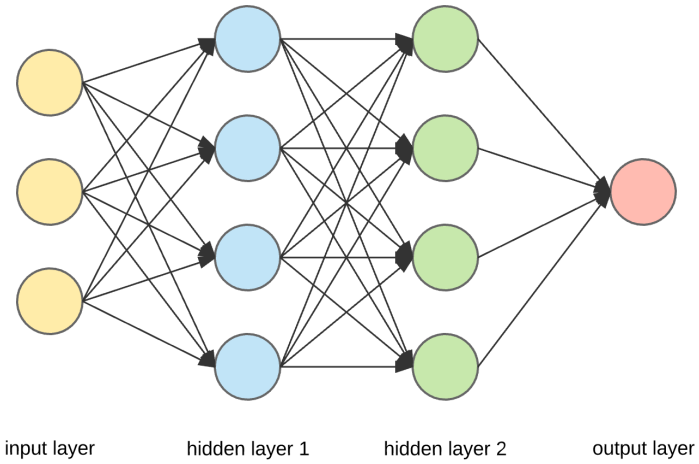

(a) Supervised machine learning.



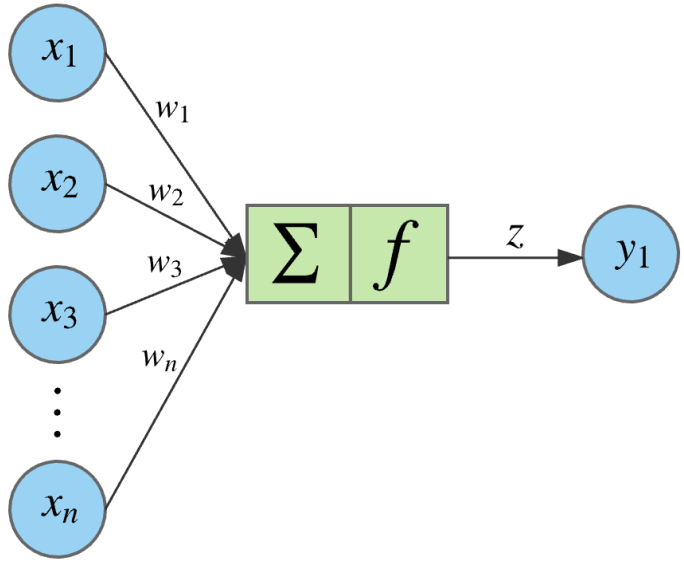

(b) Unsupervised machine learning.

Figure: Two categories of machine learning tasks. Source: `crayondata.com`

# Deep Learning




(a) Artificial neural network architecture.




(b) Neuron computation.

Figure: Artificial neural networks. Source: `miro.medium.com`

# Neural Network Training

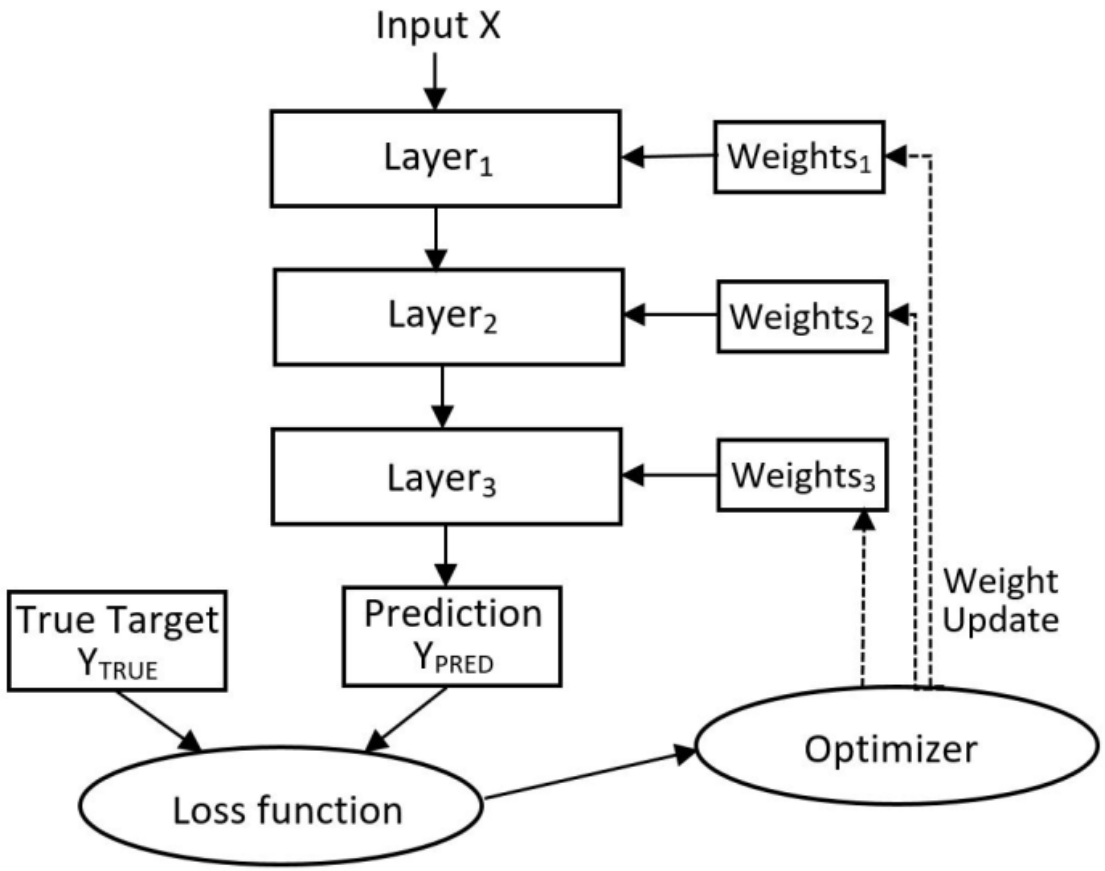



Figure: Training focuses on finding the optimal weights of each layer that best map the input instances to their corresponding targets. Source: M.R. Shahid.

# Autoencoders (AE)

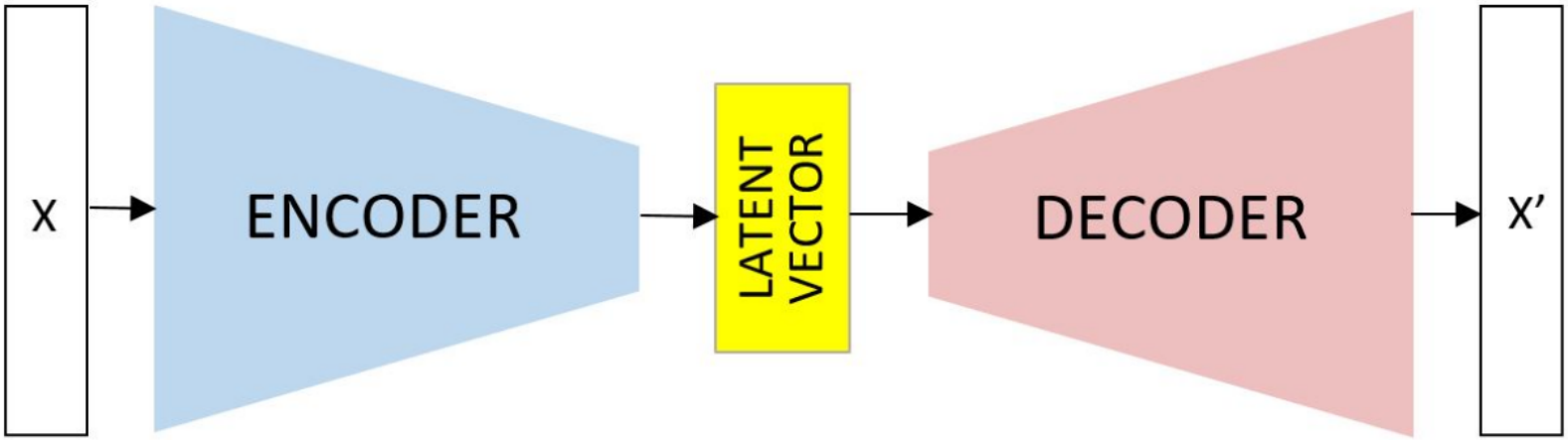



Figure: AEs are unsupervised NNs that learn to copy their inputs to their outputs under some constraints. Source: M.R. Shahid.

# Generative Adversarial Networks (GAN)

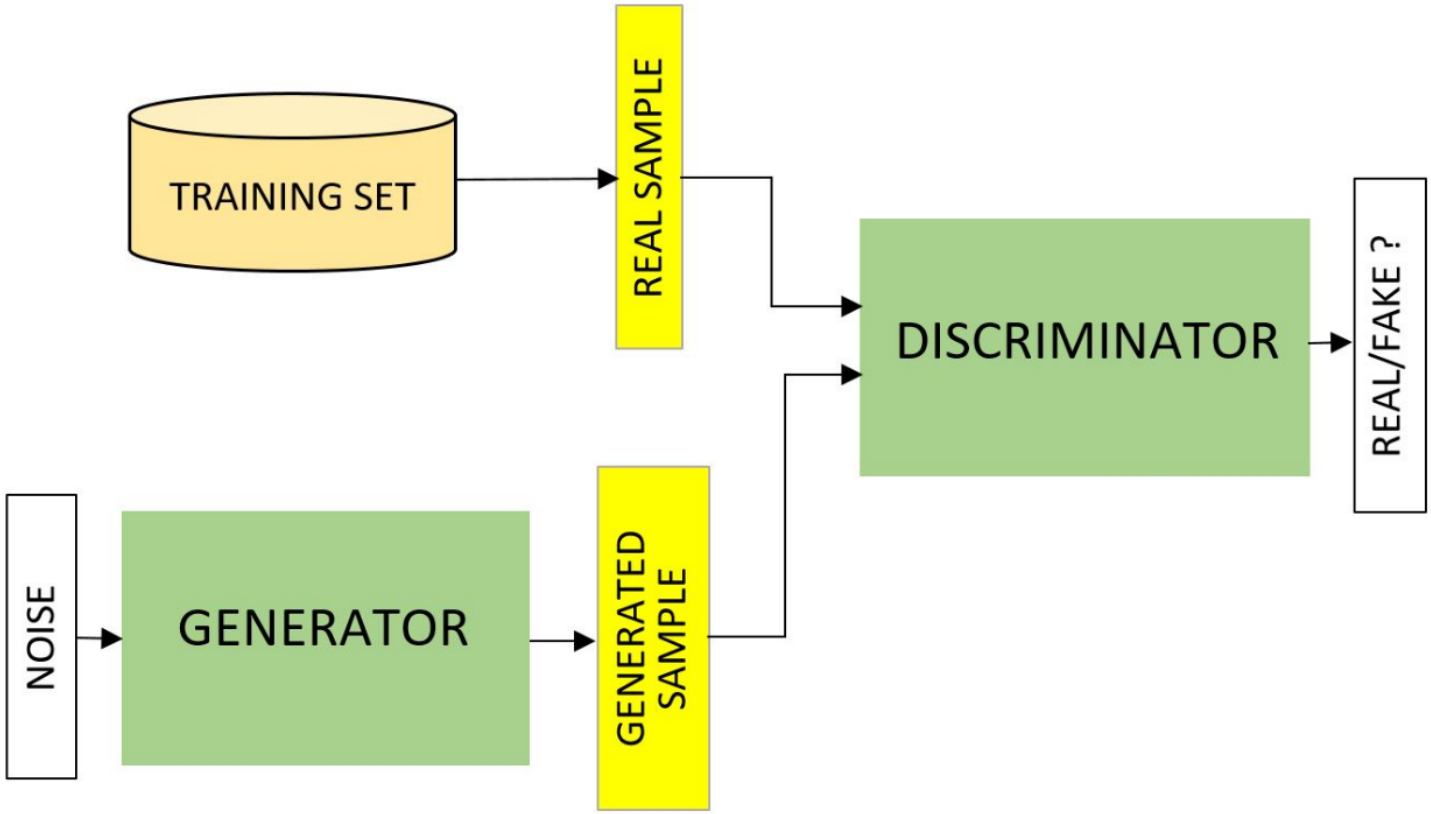



Figure: GANs are composed of two competing NNs. Source: M.R. Shahid.

# Outline

# Issues in Testing IDS

Back in 2003, NIST identified several challenges:

- difficulties in collecting attack scripts and victim software
- differing requirements for testing signature based vs. anomaly based IDS
- differing requirements for testing network based vs. host based IDS
- approaches to using background traffic in IDS tests:
  - no background traffic/logs
  - real traffic/logs
  - sanitized traffic/logs
  - generating traffic on a testbed network

**source:** Mell et al., *An Overview of Issues in Testing Intrusion Detection Systems*, NISTIR 7007, 2003

# Evaluation Metrics

According to (Milenkovski et al., 2015), IDS evaluation best practices measure (w.r.t. *attack detection*):

- **Attack detection accuracy**: *accuracy* of an IDS in the presence of *mixed workloads*
- **Attack coverage**: *accuracy* of an IDS in the presence of *pure malicious workloads*
- **Resistance to evasion techniques**:
  - *overlooked* in comparison to above two, as (Sommer & Paxson, 2010) consider it to be of limited importance from a practical perspective
  - involves *pure malicious* and *mixed* workloads
- Attack detection and reporting speed: relevant for distributed IDS

Other measurements address performance properties of IDS.

**source:** Milenkovski et al., *Evaluating Computer Intrusion Detection Systems: A Survey of Common Practices*, ACM ComSur, 2015

# Classification Metrics

Evaluating an IDS is often considered a binary classification problem. Leveraging the confusion matrix, we can measure:

- **Accuracy**: $\frac{TN+TP}{TP+FP+TN+FN)}$ (overall success rate)
- **Precision**: $\frac{TP}{TP+FP}$ (aka *positive predicted value*)
- **Detection Rate**: $\frac{TP}{TP+FN}$ (aka *sensitivity* or *recall*)
- **True Negative Rate**: $\frac{TN}{TN+FP}$ (aka *specificity*)
- **False Positive Rate**: $\frac{FP}{FP+TN} = 1 - TNR$ (aka *fall-out*)
- **F-measure**: $2 \times \frac{precision \times recall}{precision+recall}$
- **Receiver Operating Characteristic curve**: plot of the *sensitivity* as a function of $1-$ *specificity*

**source:** Moustafa et al., *A holistic review of Network Anomaly Detection Systems: A comprehensive survey*, Elsevier JNCA, 2019

# SoTA of the Evaluation of ML/DL-based IDS

Evaluation of an IDS requires:

- a testing environment
- a dataset
- a set of metrics

Evaluation methodologies usually focus on:

- dataset quality
- detection performance metrics
- realistic environment provision

# Shortcomings

Most ML/DL-based IDS proposals:

- share the same set of metrics
  - **accuracy** instead of *precision* and *recall*
  - fail to use *MCC* when the dataset is **imbalanced**
- use widespread IDS datasets
  - **KDD99** has been over-used
  - D'Hooge et al. demonstrated that many datasets suffer from **shortcut learning**
- propose comparisons
  - experimental protocols differ, e.g., **tasks are different** (supervised classification vs. anomaly detection)
  - experimental settings differ, e.g., same datasets but **different splits**

**source:** D'Hooge et al., *Establishing the Contaminating Effect of Metadata Feature Inclusion in Machine-Learned Network Intrusion Detection Models*, DIMVA'22

# Outline

# Concept Drift

Proposed NIDSs assume that the distribution of data is **stationary**. But:

- not all categories of malicious behaviour are represented uniformly across the training set
- well-established traffic features may exhibit a very gradual drift as the user habits change

Andresini et al. outline a few solutions:

- identify which characteristics change and tune the NIDS to traces exhibiting such changes
- train DNN models on historical labeled data and update them to fit unlabeled traces via transfer learning
- past models may be structurally extended to incorporate new model branches

**source:** Andresini et al., *INSOMNIA: Towards Concept-Drift Robustness in Network Intrusion Detection*, AISec'21

# Adversarial Examples

Malicious samples can be rendered **evasive** by intentionally adding **small perturbations** leading a *trained model* to misclassify them:

$$x' = x + \delta$$

Yang et al. have proposed 3 approaches to generate black-box adversarial examples:

- solving an optimization problem on a white-box substitute model and then leverage *transferability*
- estimating the gradient information to generate adequate perturbations
- training a GAN

**source:** Yang et al., *Adversarial Examples Against the Deep Learning Based Network Intrusion Detection Systems*, MILCOM'18

# Improving Intrusion Detection using GAN (1/2)

In this work, we took the angle of an adversary trying to mimic legitimate traffic, which prompted several issues:

- what is an anomaly in such context?
- what is the meaning of adversarial traffic?
- can we make the detector more robust?

Our approach was to propose a **double-objective** GAN (NOVGAN) that leverages GAN's *sword-and-shield* approach so as to **evade** IDSes:

- to generate traffic features that resemble real (legitimate) traffic
- to generate traffic that is harmful to a target network

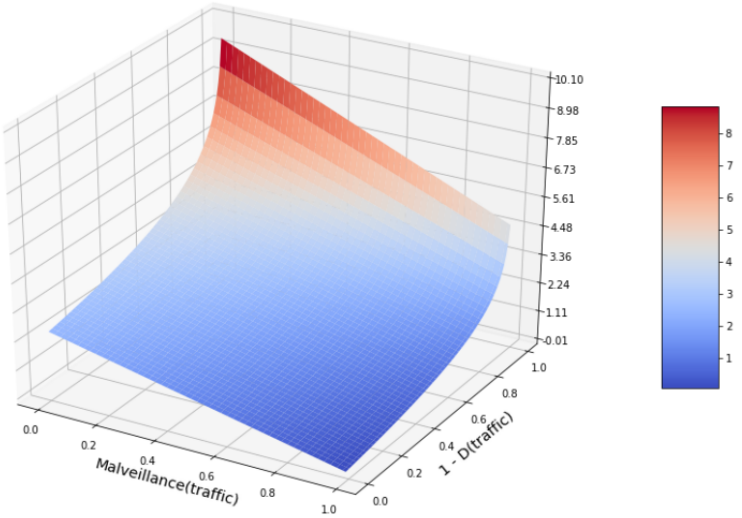
New loss function for the Generator:

$$\text{with } L_{D,G}(z) = -\log DG(z)$$

$$loss_G = \mathop{\mathbb{E}}_{z \sim P_Z}[MG(z)L_{D,G}(z) + (1 - MG(z))(\alpha L_{D,G}(z) + offset)]$$

# Improving Intrusion Detection using GAN (2/2)

Proposed loss function visualization:




Some results on NSL-KDD dataset using the *M* function as a classifier:

| Algo | Accuracy | Precision | F1-score |
|---|---|---|---|
| naive | 0.91 | 0.89 | 0.90 |
| RForest | 0.996 | 0.998 | 0.99 |

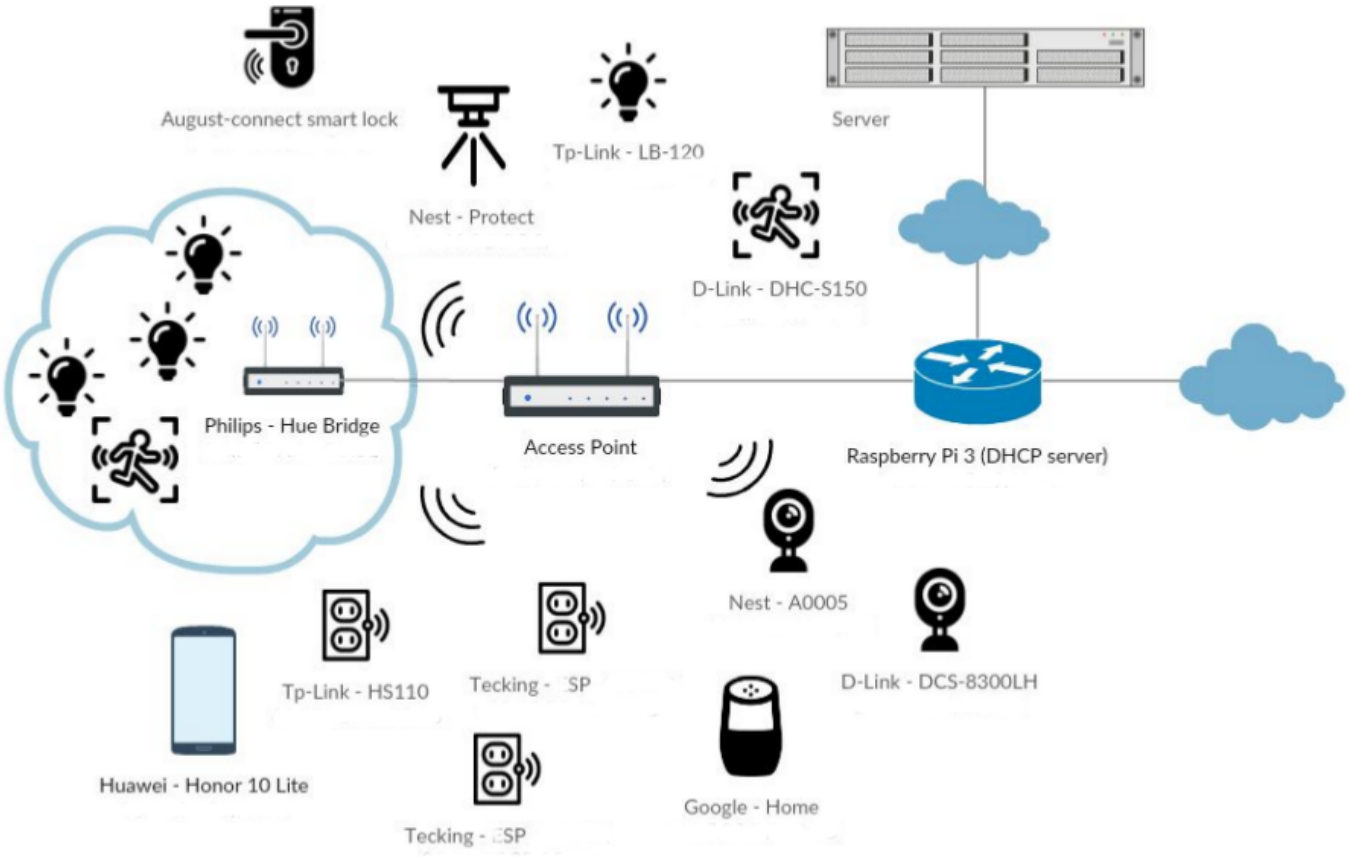
Open problems include cost reduction of parameters tuning, feature-to-traffic transformation.

**source:** Peseux, Blanc and Kiennert, *NOVGAN (draft)*, 2020

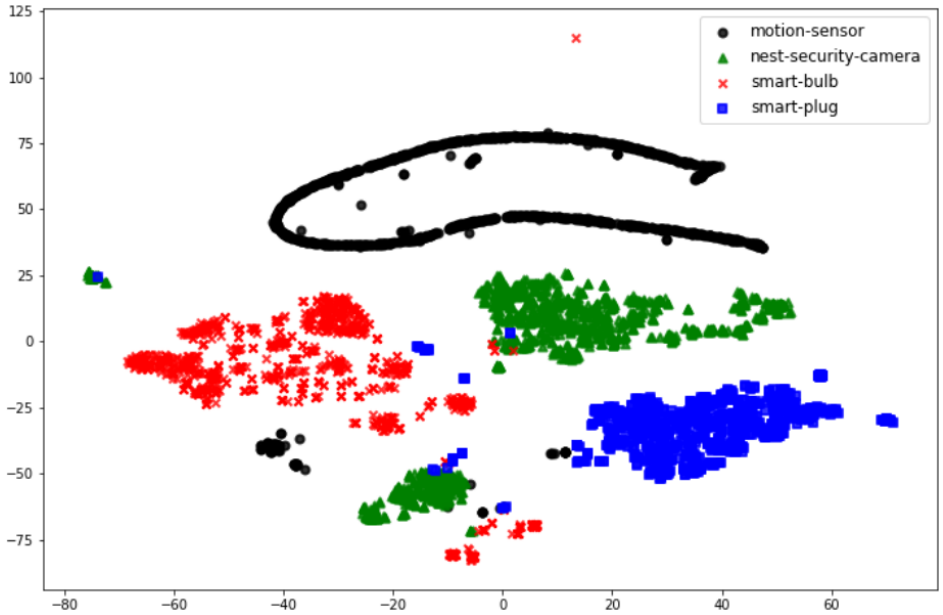
# Outline

# IoT Testbed in VARIoT

In VARIoT, we propose to generate traffic datasets and anomaly detection models for a range of IoT devices.



# IoT Identification by network traffic analysis



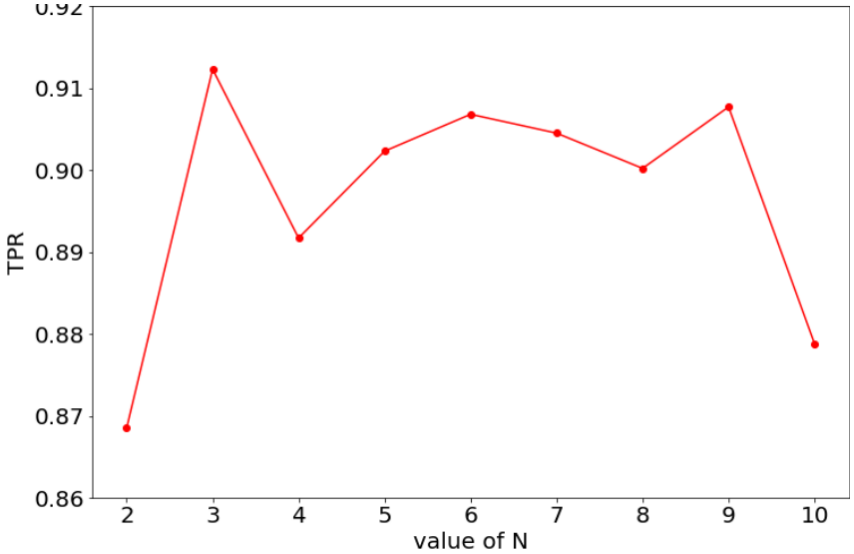

| | sensor | camera | bulb | plug |
|---|---|---|---|---|
| **RF** | 1. | 1. | .997 | .997 |
| **DT** | .993 | .995 | .995 | .997 |
| **SVM** | .997 | .988 | .997 | .984 |
| **KNN** | .995 | .988 | .986 | .984 |
| **ANN** | .990 | .986 | .989 | .978 |
| **GNB** | .985 | .871 | .880 | .958 |

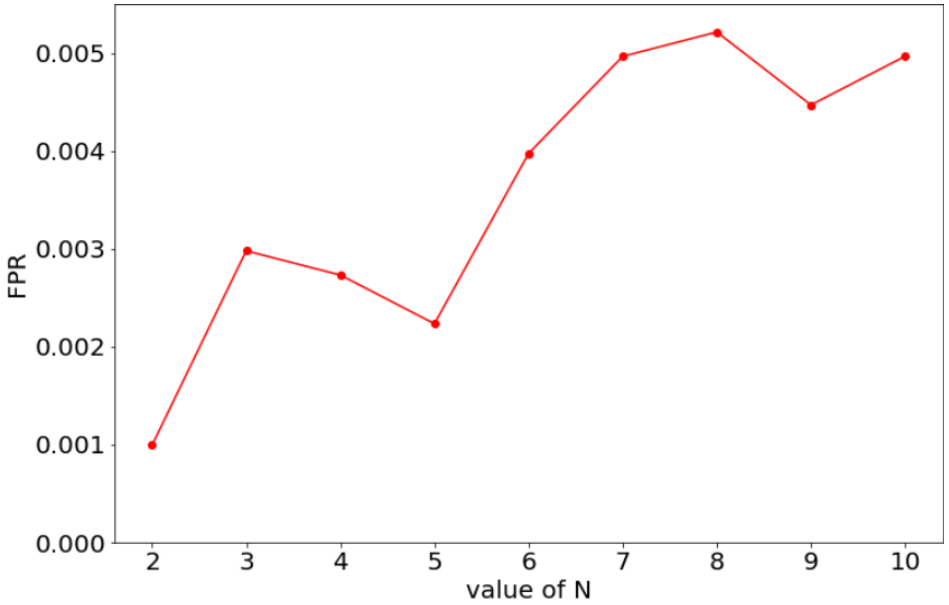
**source:** Shahid et al., *IoT Devices Recognition through Network Traffic Analysis*, Proc. of BigData'18

# Detection of anomalous IoT communications

| Features |
|---|
| Mean, Median, Min, Max, Standard deviation and Count of the size of the first N packets sent |
| Mean, Median, Min, Max, Standard deviation and Count of the size of the first N packets received |
| Mean and Standard deviation of the IAT between the first N packets sent |
| Mean and Standard deviation of the IAT between the first N packets received |

| | Bidirectional flows |
|---|---|
| D-Link Motion Sensor | 1074 |
| Nest Security Camera | 1055 |
| TP-Link Smart Bulb | 1040 |
| TP-Link Smart Plug | 858 |
| **Total** | **4027** |




**source:** Shahid et al., *Anomalous Communications Detection in IoT Networks using Sparse Autoencoders*, NCA'19

# Outline

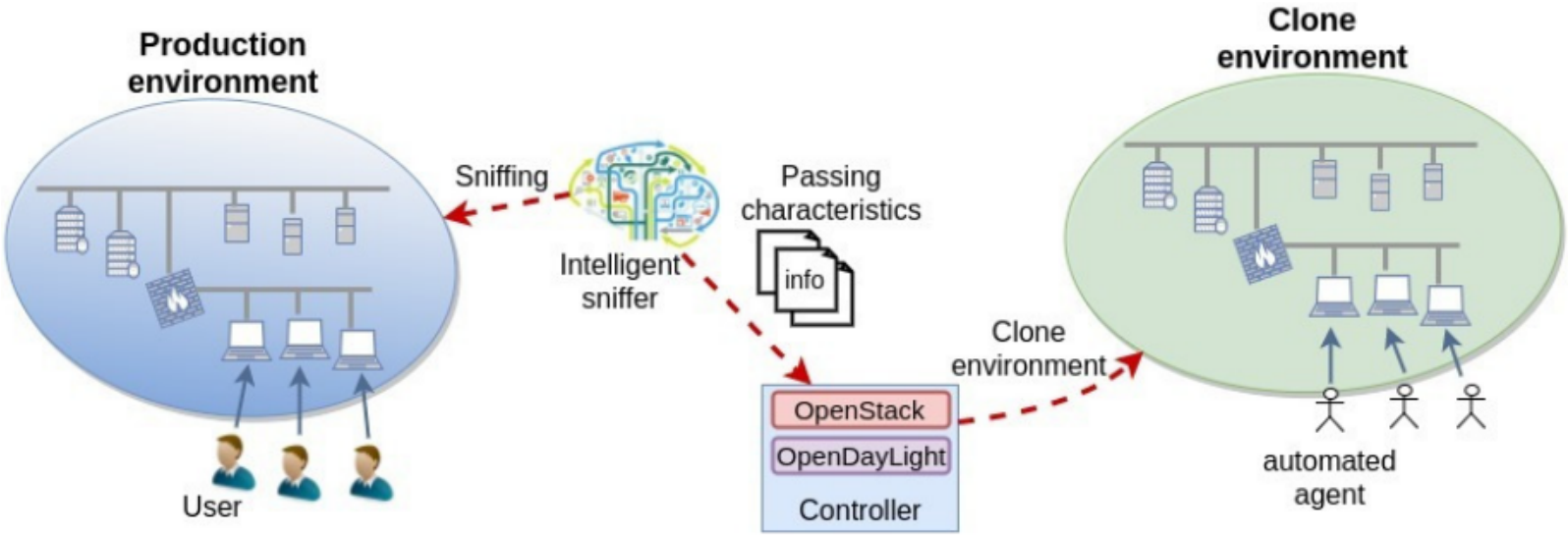
# Network Intrusion Detection Datasets (Ring et al., 2019)

- Labeled and representation network-based datasets are necessary to compare the quality of different NIDS
- Few labeled datasets are publicly available
- Available datasets are often outdated
- Using real network traffic is also problematic due to the missing ground truth (manual labeling is difficult)
- Real network traffic cannot often be shared due to privacy concerns

**Proposal:** generate realistic flow-based network traffic

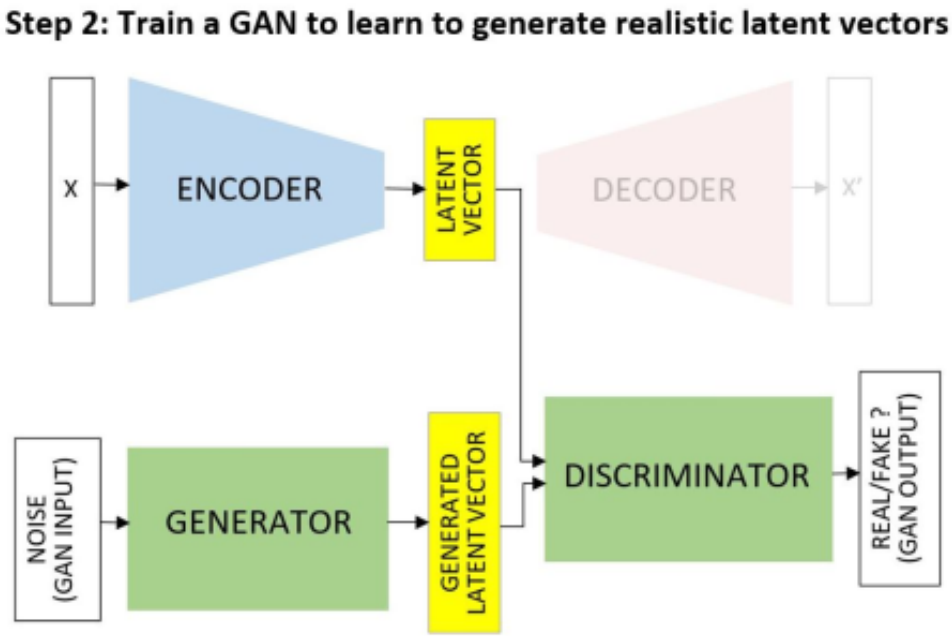
**source:** Ring et al., *Flow-based Network Traffic Generation using Generative Adversarial Networks*, Elsevier Computers & Security, 2019

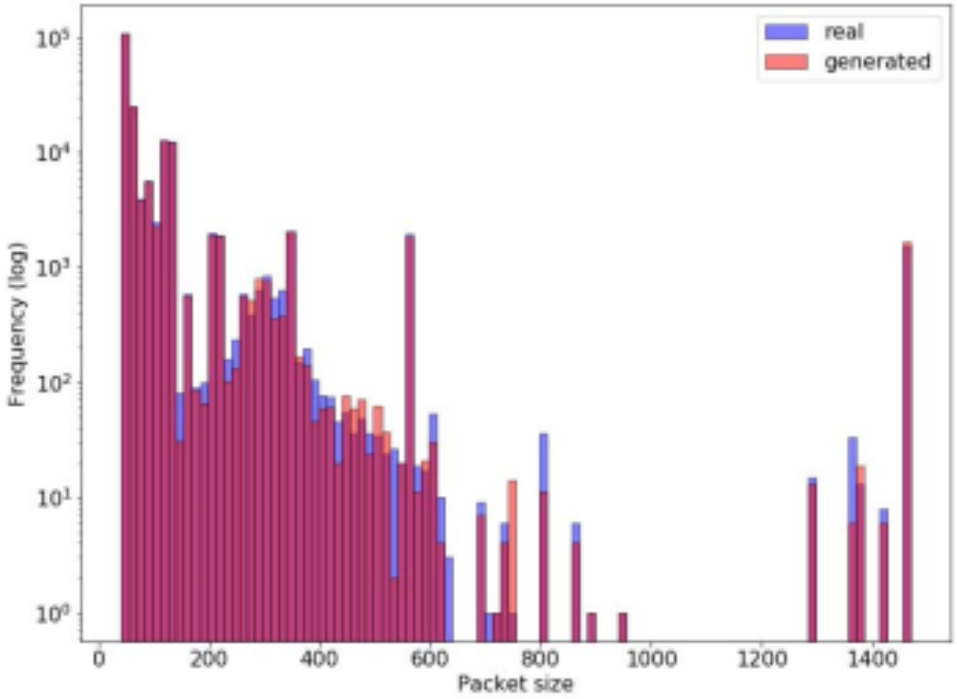
# Statistical Legitimate Traffic Generation




**source:** Pham et al., *On Automatic Network Environment Cloning for Facilitating Cybersecurity Training and Testing*, RESSI'18

# Learning-based IoT Traffic Generation




**source:** Shahid et al., *Generative Deep Learning for Internet of Things Network Traffic Generation*, PRDC'20

# Feature Space vs. Problem Space

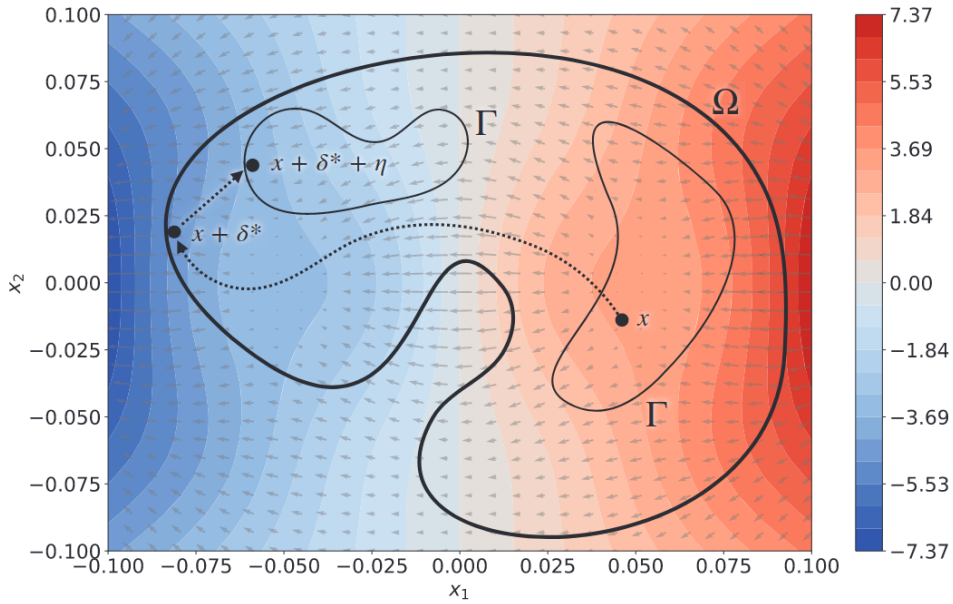



Figure: Example of projection of the feature-space attack vector $x + \delta*$ in the *feasible* problem space, resulting in side-effect features $\eta$

**source:** Pierazzi, Pendlebury et al., *Intriguing Properties of Adversarial ML Attacks in the Problem Space*, S&P'20

# Evaluating a Generator

Dataset, although synthetic, still requires a certain level of quality. Since no generally applicable evaluation method was available, we propose our criteria:

- **Realism**: a synthetic sample should be sampled from the same distribution as the real data
- **Diversity**: the distribution of the generated samples should have the same variability as the real data
- **Originality**: a generated sample should be sufficiently different from the samples of the real distribution
- **Compliance***: generated network traffic must also conform to specifications, standards

**source:** Schoen et al., *Towards generic quality assessment of synthetic traffic for evaluating intrusion detection systems*, RESSI'22

# Towards Traffic Morphing

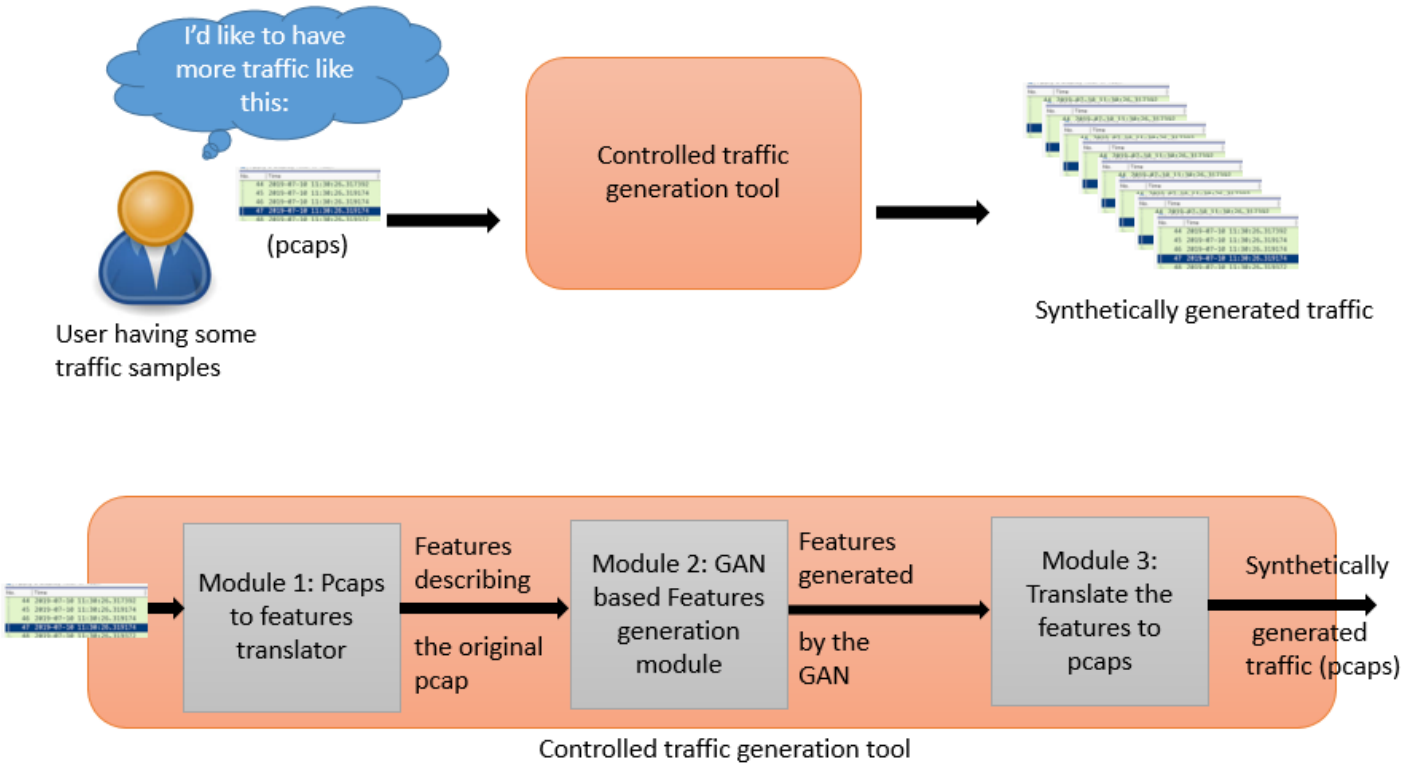



Figure: Synthetic traffic generation

# Future works

- in the feature space, realistic diverse and compliant generation of NetFlow flows, network packets and application payloads
- in the feature space, data-driven exploration of adversarial space
- in the problem space, generative network based generation of attacks using tools
- from feature to problem space, exploitation of traffic morphing
- development evaluation and explainability methodologies

## Thank you for your attention!


https://cloudgravity.github.io


@cloudgravity


gregory.blanc@telecom-sudparis.eu


https://anr-grifin.telecom-sudparis.eu/