Robust URL Classification With Generative Adversarial Networks

Martino Trevisan

Idilio Drago

URL classification – What is this?

URL classification means **accounting** a URL to a particular class.

- Video Streaming (e.g., Netflix, CNN, …)
- Software Update (Windows, Antivirus)
- Malware

Useful for:

- Content filtering
- Cyber Security
- Accounting / traffic measurements

URL classification – Example

Video Streaming

Fixed pattern
+
Variable parts (CDN nodes, timestamp,…)

```
se-rm3-14.se.live1.msp.ticdn.it/content/ss/live/channel(ch35).isml/QualityLevels(96000)/Fragments(audio_ita=1620605440099198)
se-mi1-17.se.live1.msf.ticdn.it/Content/SS/Live/Channel(CH02).isml/QualityLevels(96000)/Fragments(audio_ita=1620605420101802)
se-mi1-14.se.live1.msp.ticdn.it/content/ss/live/channel(ch35).isml/QualityLevels(1600000)/Fragments(video=1620603902765864)
se-mi1-17.se.live1.msf.ticdn.it/Content/SS/Live/Channel(CH02).isml/QualityLevels(96000)/Fragments(audio_ita=1620604600048468)
```

Malware: *Tidserv* Fast Flux Ph

Path is pseudo-random
+
Hostname not fixed

```
wuptywcj.cn/XZc2GhZD7z4ymgo3dmVyPTUuMCZzPTAmYmlkPWE0OTY1YjNlMTFlZmMmYZz13d3cuZ29vZ2xlLml0JnE9dWlzcCZ4ODY9MzI=27x
wuptywcj.cn/xKP3jVbd7g4jpjs5dmVyPTUuMCMyMzJjY2U2YWQ2OGQmYWlkPTMwNDI4JnNpZD0Lml0JnE9ZW1pbGJhbmNhJng4Nj0zMg==27x
rlyg0-6nbcv.com/Kvb13nWd6P4XrFs3dmVyPTQuZDA5N2RiYmRlYmVkZiZhaWQ9NTAwMTgmc2lkPTAmcmQ2dsZS5pdCZxPWZhY2Vib29r27c
wuptywcj.cn/KkV3WhhD7m5ZD0wJmVuZz13d3cuZ29vZ2xlLml0JnE9bmF0YWxpZSUyMGltYnJ1Z2xpYSUyMC0lMjB0b3JuJTIwdHJhZHV6
```

URL classification – Example

Video Streaming

Classical Solution:
Manually create a *Regular Expression*

```
se-rm3-14.se.live1.msp.ticdn.it/content/ss/live/channel(ch35).isml/QualityLevels(96000)/Fragments(audio_ita=1620605440099198)
se-mi1-17.se.live1.msf.ticdn.it/Content/SS/Live/Channel(CH02).isml/QualityLevels(96000)/Fragments(audio_ita=1620605420101802)
se-mi1-14.se.live1.msp.ticdn.it/content/ss/live/channel(ch35).isml/QualityLevels(1600000)/Fragments(video=1620603902765864)
se-mi1-17.se.live1.msf.ticdn.it/Content/SS/Live/Channel(CH02).isml/QualityLevels(96000)/Fragments(audio_ita=1620604600048468)
```

Malware: *Tidserv* Fast Flux Phis

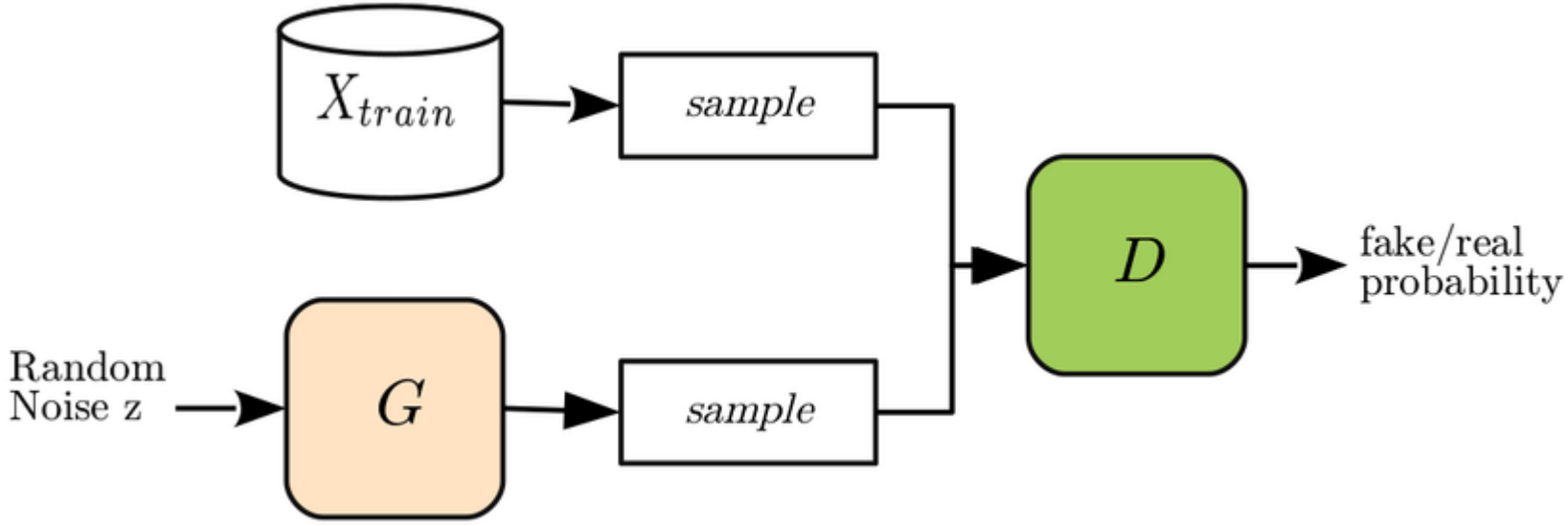
Classical Solution:
Train a classifier provided
(i) samples of the malware
(ii) Normal traffic (non-malware)

```
wuptywcj.cn/XZc2GhZD7z4ymgo3dmVyPTUuMCZzPTAmYmlkPWE0OTY1YjNlMTFlZmMmYZz13d3cuZ29vZ2xlLml0JnE9dWlzcCZ4ODY9MzI=27x
wuptywcj.cn/xKP3jVbd7g4jpjs5dmVyPTUuMCMyMzJjY2U2YWQ2OGQmYWlkPTMwNDI4JnNpZD0Lml0JnE9ZW1pbGJhbmNhJng4Nj0zMg==27x
rlyg0-6nbcv.com/Kvb13nWd6P4XrFs3dmVyPTQuZDA5N2RiYmRlYmVkZiZhaWQ9NTAwMTgmc2lkPTAmcmQ2dsZS5pdCZxPWZhY2Vib29r27c
wuptywcj.cn/KkV3WhhD7m5ZD0wJmVuZz13d3cuZ29vZ2xlLml0JnE9bmF0YWxpZSUyMGltYnJ1Z2xpYSUyMC0lMjB0b3JuJTIwdHJhZHV6
```

Generative Adversarial Networks [1]

Unsupervised Learning

- Given training samples, the model learns to generate other samples:
  - Realistic - that look like the original ones
  - But are not exactly the same



[1] Goodfellow, Ian, et al. "Generative adversarial nets." *Advances in neural information processing systems*. 2014.

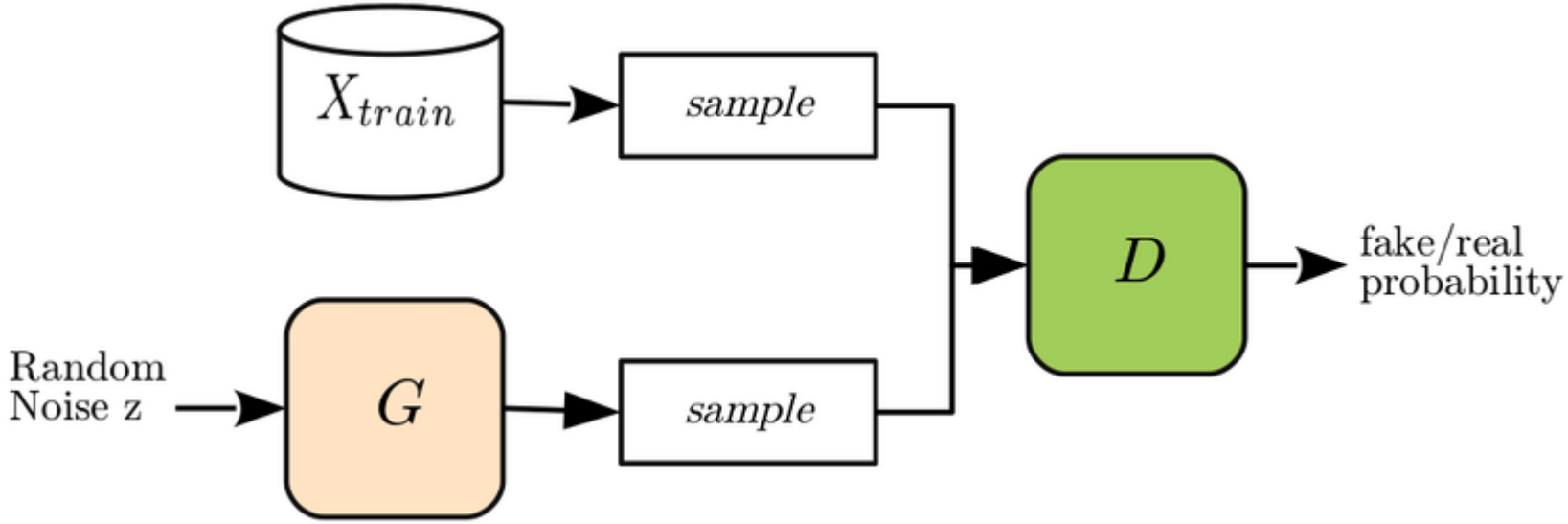
Generative Adversarial Networks [1]

Composite model:

1. Generator: take random noise => Generate realistic samples
2. Discriminator: distinguish generated from original samples

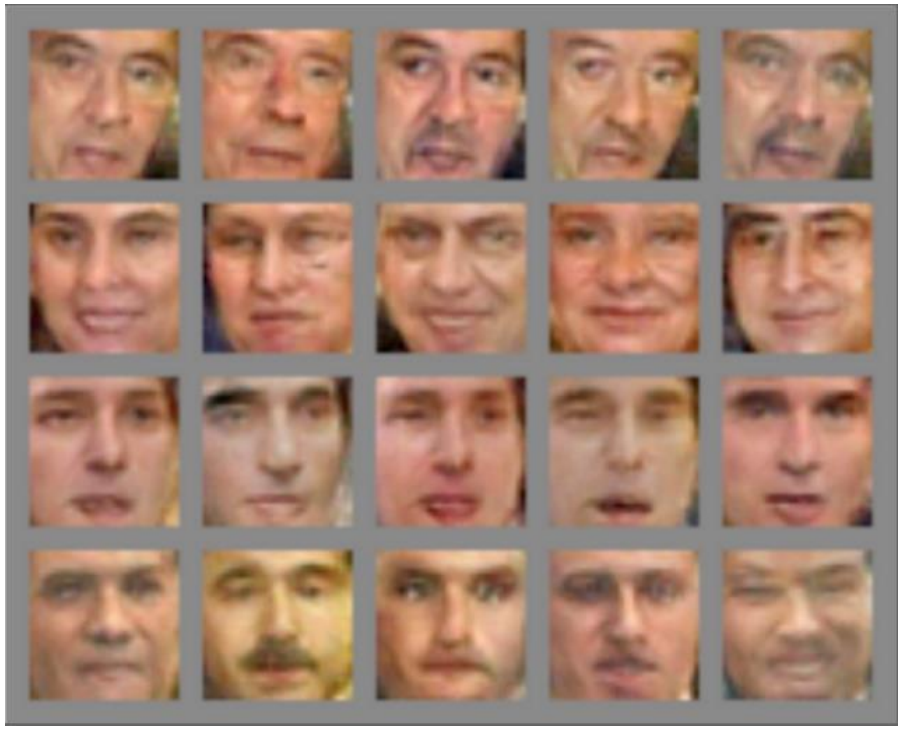
They are adversaries, in constant battle during the training process

They are often neural networks



[1] Goodfellow, Ian, et al. "Generative adversarial nets." *Advances in neural information processing systems*. 2014.

Generative Adversarial Networks



Use GANs for URL classification (and generation)

Key Idea:

- Collect example data of URL of interesting class
- Train a GAN for each class
- Use discriminators to identify new URLs in a live stream of URLs

Pros:

- No manual intervention
- No samples of other URLs / normal data

Cons:

- It is still an empiric model
- If something goes wrong, you do not know why

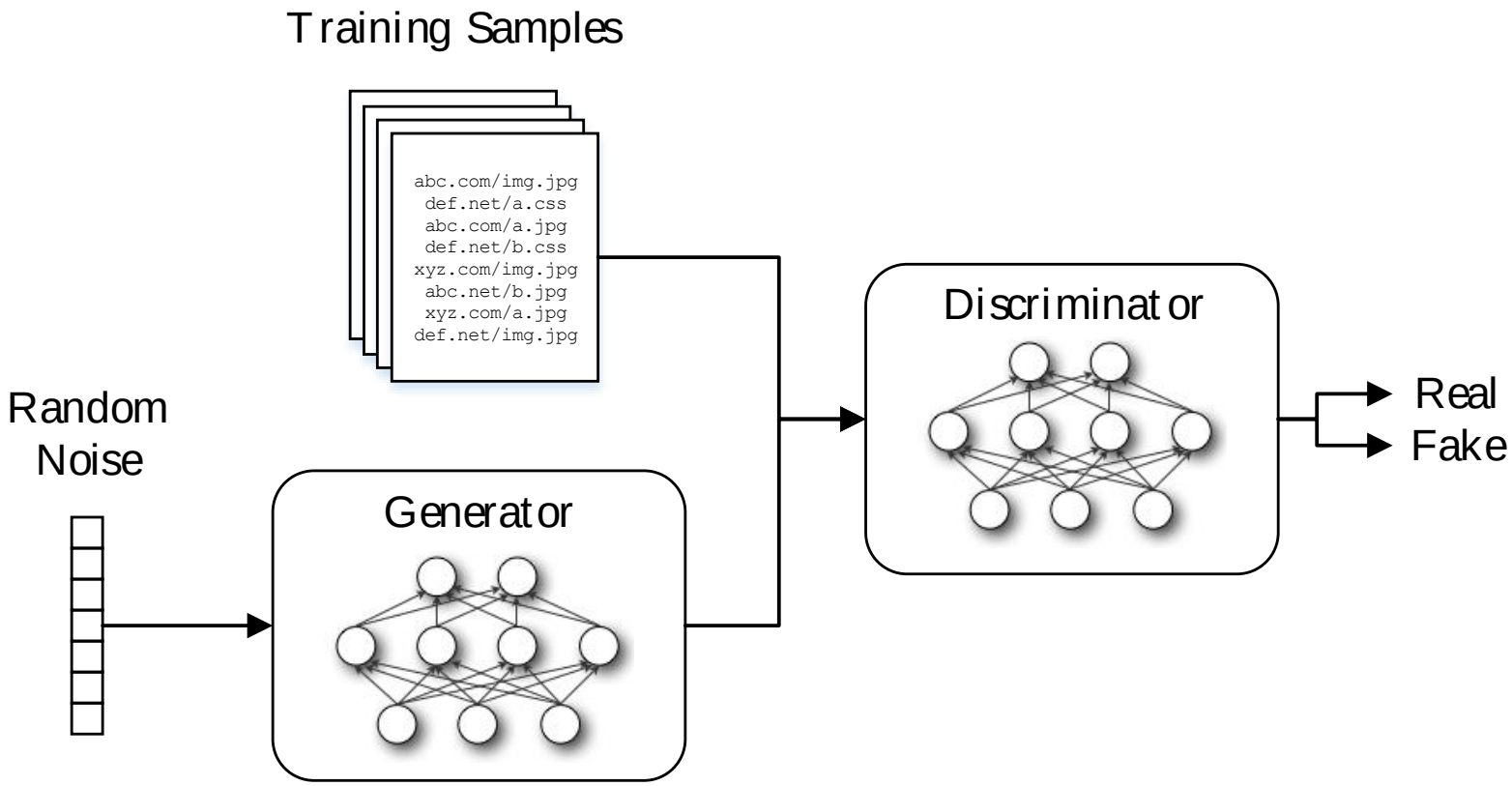
Use GANs for URL classification (and generation)

For each class of interest:

- Collect example data

1. Train a GAN – both models are simple feedforward NN
2. Use the discriminator to classify new URLs

1)



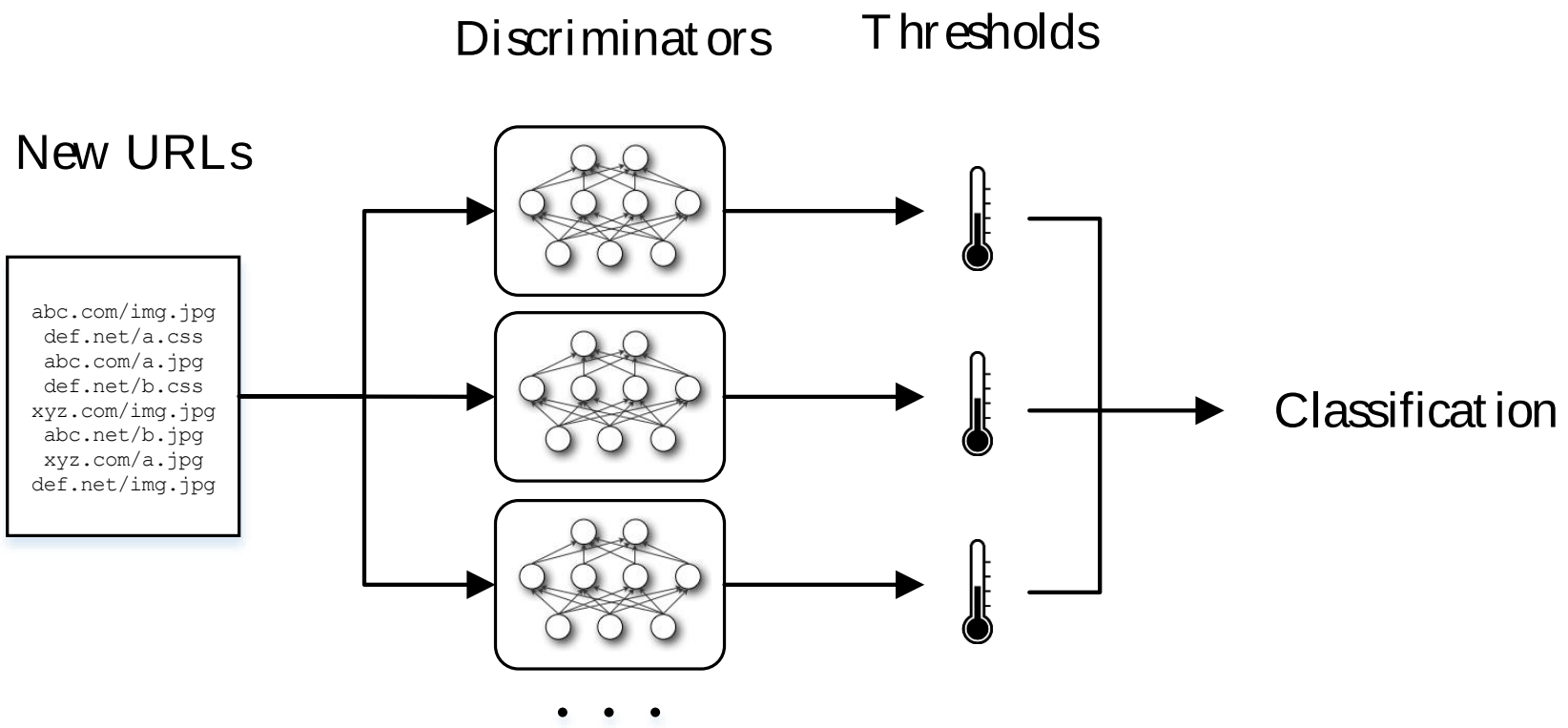
Use GANs for URL classification (and generation)

For each class of interest:

- Collect example data

1. Train a GAN
2. Use the discriminator to classify new URLs

2)



Our experiments

4 URL classes – real URLs from an operational network

Name	#URLs	Description
Video	8 620	Video Streaming chunks
Checkpoint	17 451	CheckPoint firewall updates
Windows	5 277	Windows update archives
Tidserv	227	TidServ malware

Split in:

- Training set: train 4 GANs
- Test set: used at classification time

+

Others	24 667	Other URLs

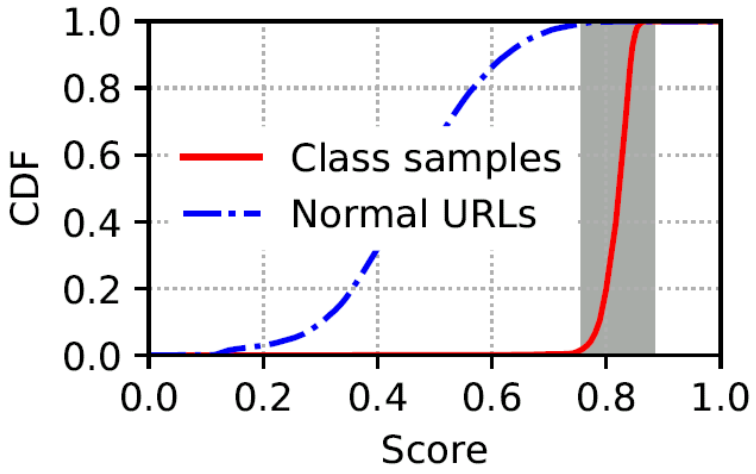
Used to quantify the ability of classification

We imagine a scenario where you have a deluge of Normal URLs, and we want to pinpoint interesting ones (e.g., content filtering, advanced accounting, …)

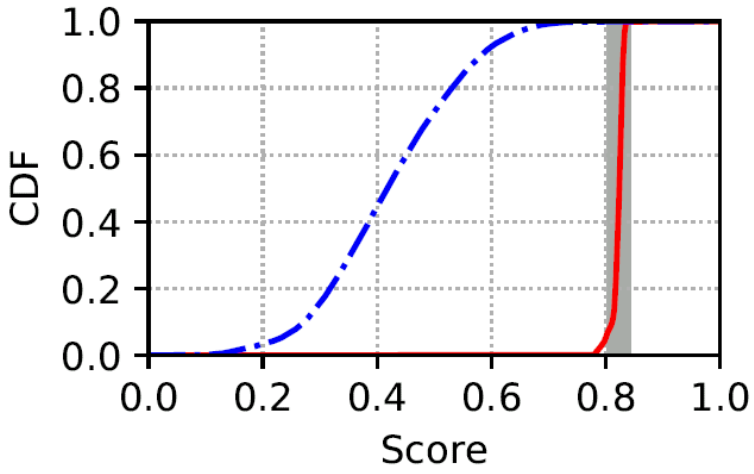
Discriminator Output

Train a GAN for each cla
By design, discriminator
- 1: for URLs belonging
- 0: for the others

It is trivial to fix a threshold!!!
We use the boxplot rule [2]



(a) *Checkpoint*

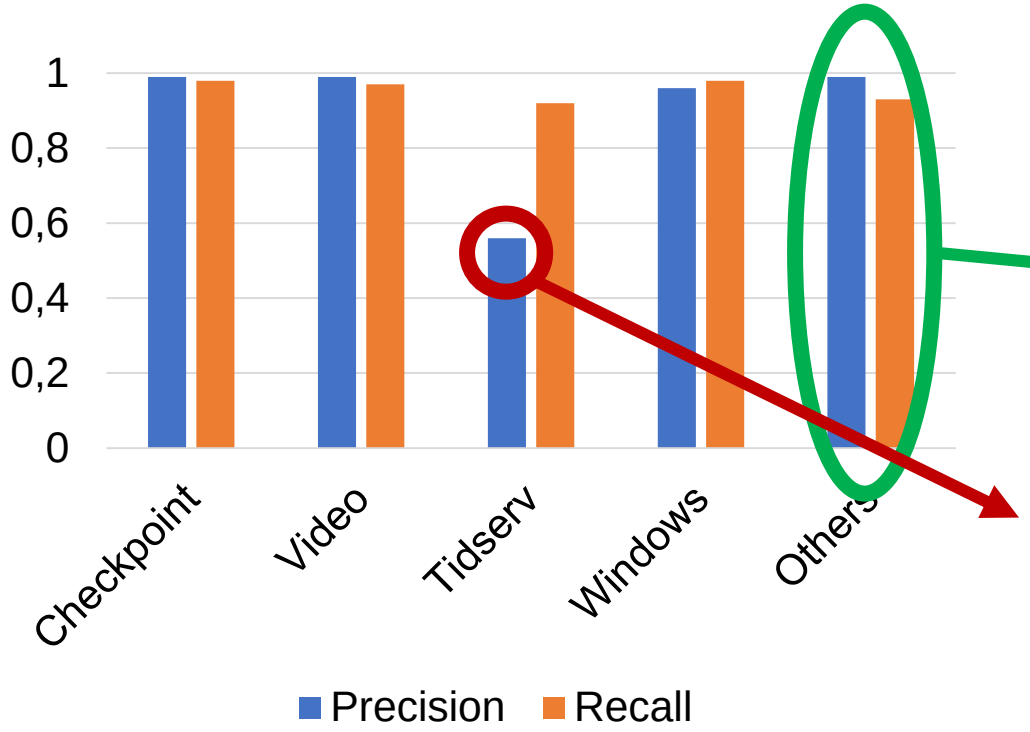


(b) *Video*

[2] Laurikkala, Jorma, et al. "Informal identification of outliers in medical data." Fifth International Workshop on Intelligent Data Analysis in Medicine and Pharmacology. Vol. 1. 2000.

Classification performance

- Take the test set URLs
- Add the Normal URLs
- Compute classical performance metrics



Precision and recall are generally high (>90%)

Not used at training time ☺

Tidserv (fast-flux malware) has high recall, but low precision ☹
That's because we have only 227 URL sample, and URLs are very variable
Note the high recall ☺

And the generators?

For each URL class, we trained a discriminator and **a generator**
How generated URLs look like?

Checkpoint
cws.checkpoint.com:80/malware/malware/6.0?resource=ew5rnhlvdS5jb20=&key=123456
cws.checkpoint.com:80/malware/malware/6.0?resource=ew91xyjlchjpb3iuy29t&key=123456

Original

cws.checkpoint.com:80/malware/malware/6.0?resource=1mq4lcktllu41;hpkiquy254bvrbqrz_22
cws.checkpoint.com:80/malware/malware/6.0?resource=mgyu/oiuz25o-z0lr=2yf2=bw1u0ij

Generated

Video
hsslive.pcdn.any.sky.it/21920/sport1.isml/qualitylevels(850000)/fragments(video=926861730399413)
hsslive.pcdn.any.sky.it/22362/tg24go.isml/qualitylevels(918000)/fragments(video=2367722273934346)

Original

hsslive.pcdn.any.sky.it/21920/sport1.isml/qualitylevels(69000)0frag7gmjsta(dio.o=a=626ig6030266t)
hsslive.pcdn.any.sky.it/21920/sport1.isml/qualitylevels(140000/frfgmentn(auvid_ot9=6868v7f3090b0))

Generated

High-level patterns are correctly learned ☺
URL semantic is not respected ☹

In conclusion

GANs are a promising technique for unsupervised learning, used mostly for image processing

We show that can be used also with network data (URLs in this case)

- Good for classification
- Promising for generation

Still preliminary work:

- More Classes, more URLs
- Comparison with other approaches
- Scalability?

Possible fields of application in networking

- Privacy: automatically identify ads/tracker URLs
- Cyber security: generate / identify attacks
- Network emulation: realistic 3G/4G emulation