State of the Art (2021)

Deep Learning for Cybersecurity

For full list of references visit: https://bit.ly/3x5DycJ

(Amar Meddahi 2021)

Outline

- Neural Networks and Deep Learning
  - Overview
  - Generative Adversarial Networks
- Generative Adversarial Networks for Network Security
  - Literature Review
  - Network Traffic Generation
- Q&A

For full list of references visit: https://bit.ly/3x5DycJ

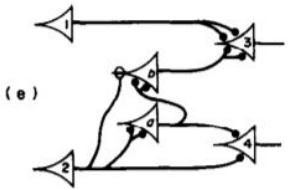
(Amar Meddahi 2021)

Neural Networks and Deep Learning

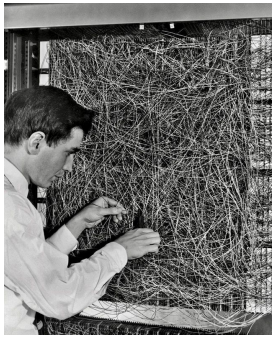
For full list of references visit: https://bit.ly/3x5DycJ

(Amar Meddahi 2021)

Overview

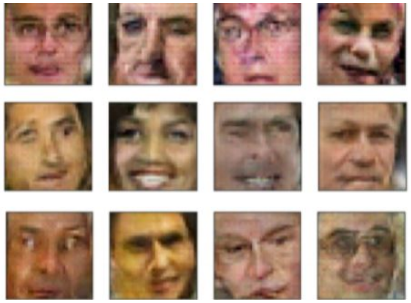


1943 : McCulloch & Pitts Publish One of the First Studies of Pattern Recognition [1]



1960 : Frank Rosenblatt with a Mark I Perceptron computer [2]

1989: **Y. LeCun et al.** *Backpropagation Applied to Handwritten Zip Code Recognition* [3]

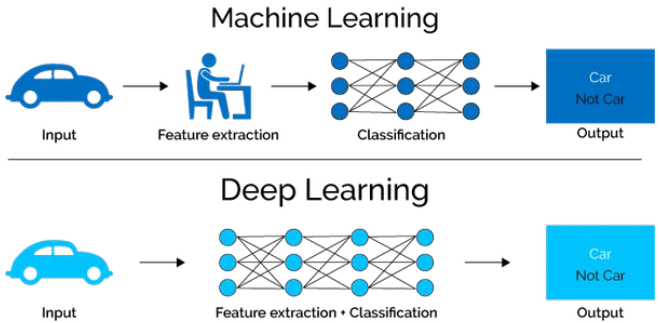


2014: Generative Adversarial Networks [4]

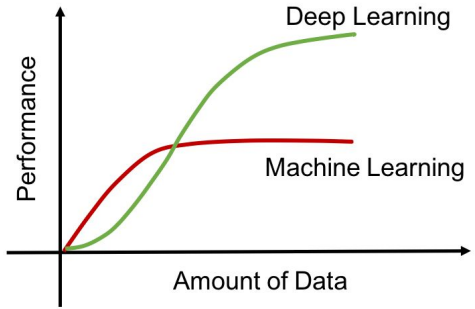
For full list of references visit: https://bit.ly/3x5DycJ

(Amar Meddahi 2021)

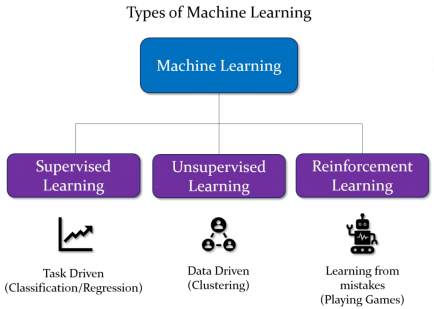
Overview



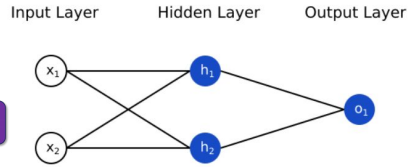
Remove the human from the picture



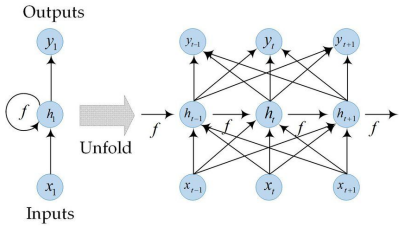
The explosion of available data



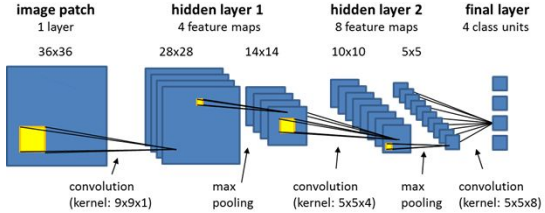
Learning



Standard Neural Networks



Recurrent Neural Networks



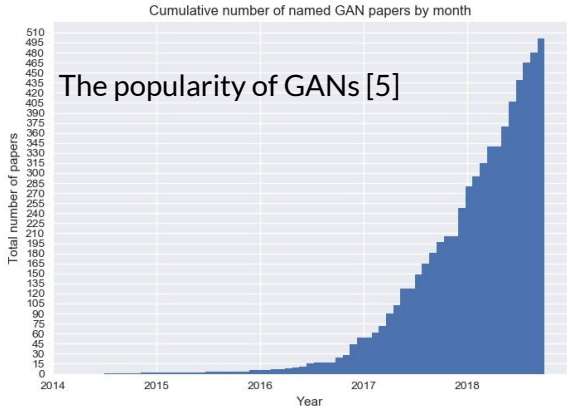
Convolutional Neural Networks

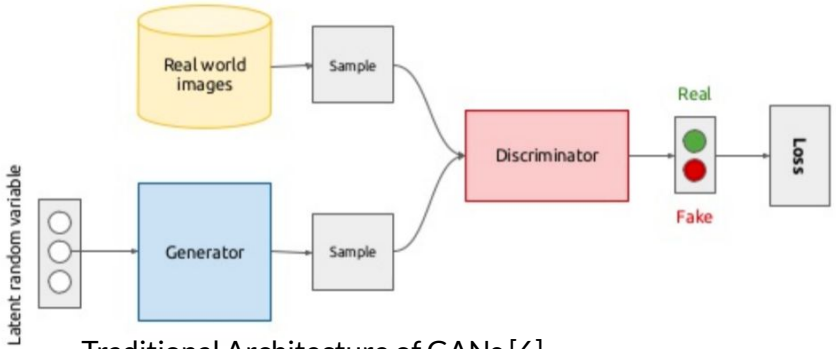
For full list of references visit: https://bit.ly/3x5DycJ

(Amar Meddahi 2021)

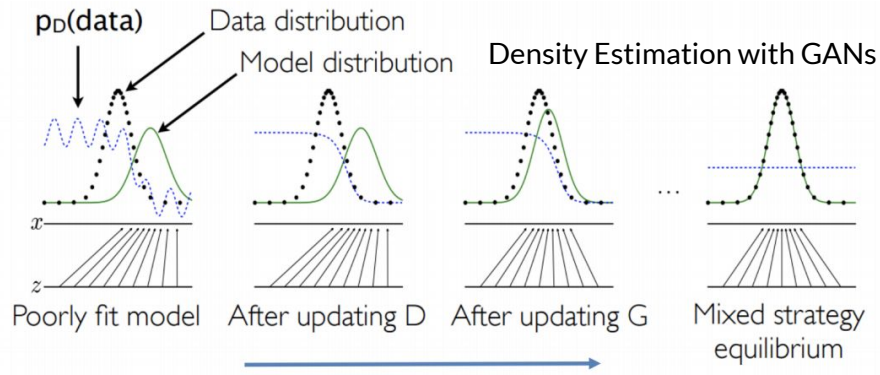
Generative Adversarial Networks (GANs)

Ian Goodfellow : First contributor of GANs (2014) [4]





Traditional Architecture of GANs [6]



For full list of references visit: https://bit.ly/3x5DycJ

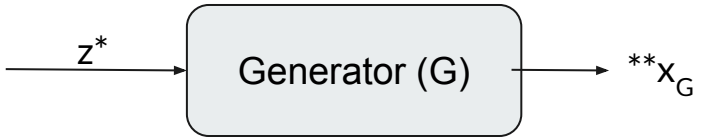
(Amar Meddahi 2021)

Technical and Theoretical Considerations

*(real data or a sample from the generator)

Cost function of the discriminator (cross entropy)

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{x\sim p_{data}} log(D(x)) - \frac{1}{2}\mathbb{E}_{z} log(1 - D(G(z)))$$



*(random noise) **(sample)

Cost function of the generator

$$J^{(G)} = -J^{(D)}$$

We therefore look for the optimal parameters $\theta^{(G)*}$ of a generator that verifies the following minimax equation:

$$\theta^{(G)*} = \arg\min_{\theta^{(G)}} \max_{\theta^{(D)}} -J^{(D)}$$

The solution to this problem is a Nash equilibrium ($\theta^{(D)}$, $\theta^{(G)}$), where :

- $\theta^{(G)}$ designates the parameters of a generator such as : G(z) ~p_data
- $\theta^{(D)}$ designates the parameters of a discriminator such as : D(x) = ½ i.e. the generator samples are indistinguishable from the genuine samples.

For full list of references visit: https://bit.ly/3x5DycJ

(Amar Meddahi 2021)

Technical and Theoretical Considerations

From a practical point of view, there is no algorithm that converges to this equilibrium point. In fact, a simultaneous gradient descent is used, without guaranteeing a solution from a theoretical perspective. I. Goodfellow et al. describes the learning phase as follows

Algorithm 1: Simultaneous gradient descent for GAN training

for $n = 1$ **to** N_{iter} **do**

–Discriminator training–

Sample minibatch of m noise samples $(z^{(1)},...,z^{(m)})$

Sample minibatch of m examples $(x^{(1)},...,x^{(m)})$

Update the discriminator by ascending its stochastic gradient (i.e. to maximize the loss):

$$\frac{1}{m}\sum_{i=1}^{m} logD(x^{(i)}) + log(1 - D(G(z^{(i)})))$$

–Generator training–

Sample minibatch of m noise samples $(z^{(1)},...,z^{(m)})$

Update the generator by descending its stochastic gradient (i.e. to minimize the loss):

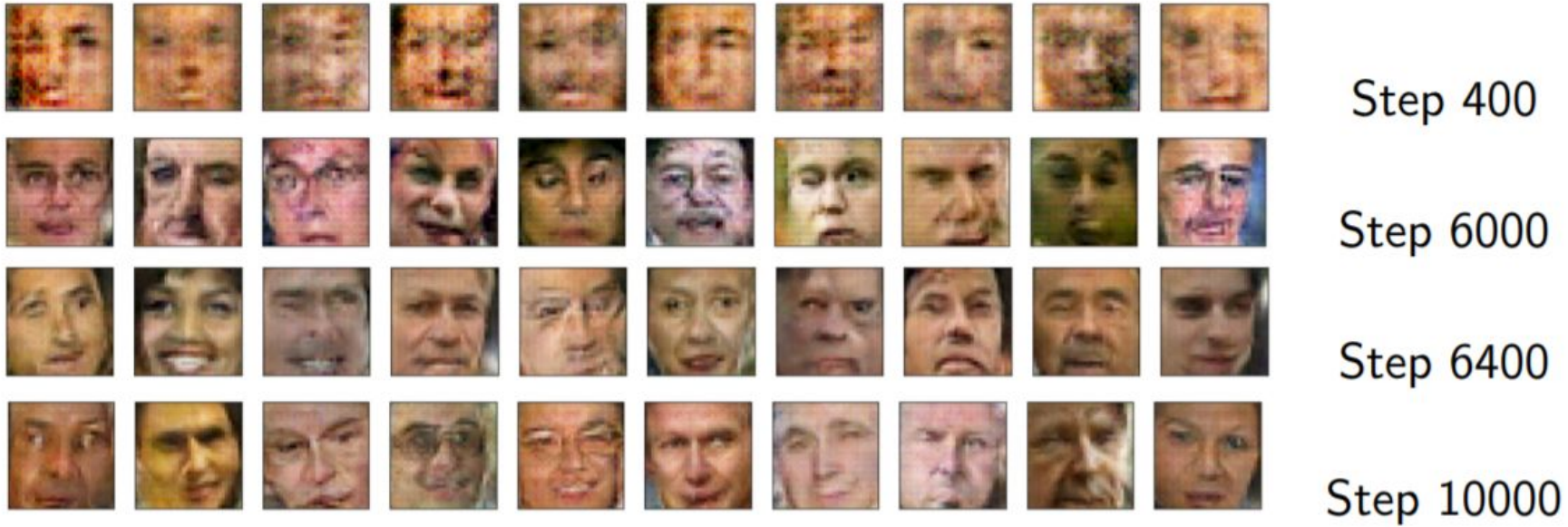
$$\frac{1}{m}\sum_{i=1}^{m} log(1 - D(G(z^{(i)})))$$

end

For full list of references visit: https://bit.ly/3x5DycJ

(Amar Meddahi 2021)

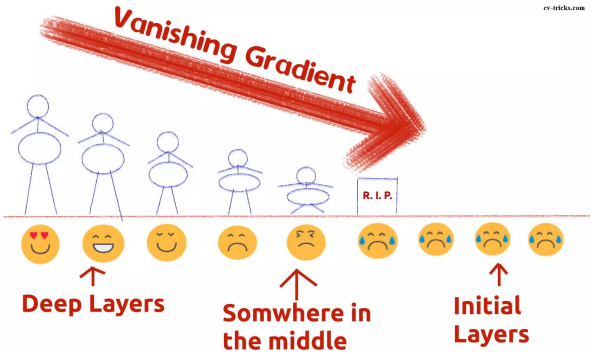
Some results from the literature



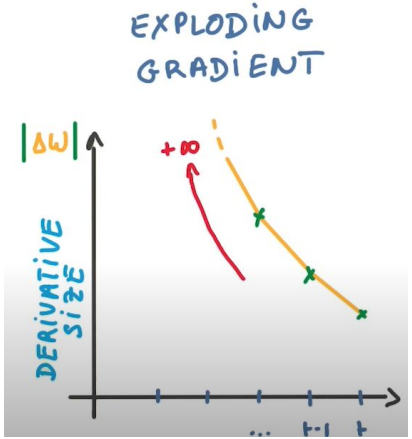
For full list of references visit: https://bit.ly/3x5DycJ

(Amar Meddahi 2021)

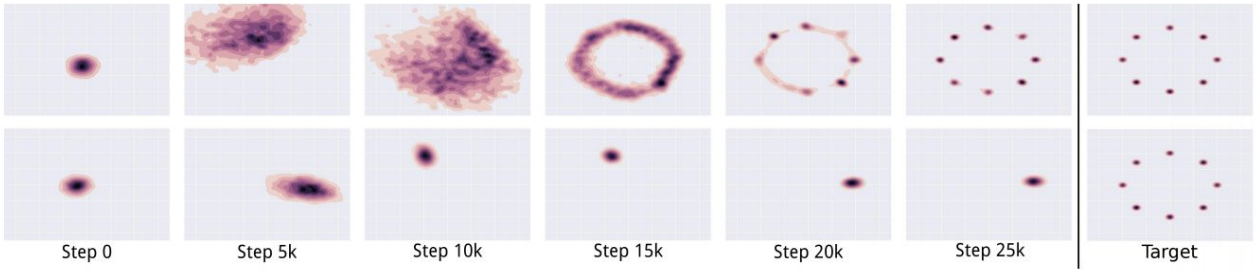
General problems of GANs



Vanishing gradient problem [7]



Exploding gradient [8]

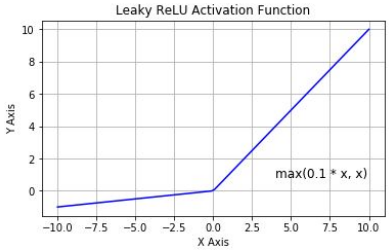


Mode collapse [9]

For full list of references visit: https://bit.ly/3x5DycJ

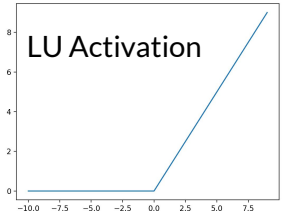
(Amar Meddahi 2021)

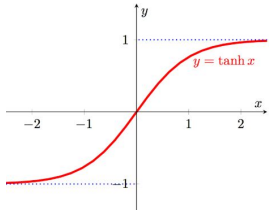
DCGAN (2016)

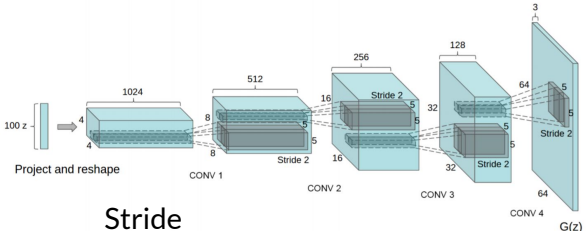


The Deep Convolutional Generative Adversarial Network (DCGAN) [10] article is fundamental because it intr a series of recommendations for the architecture of the generator and the discriminator:

- **Do not use a pooling layer, but rather stride**
- **Use ReLU activations for the hidden layers of the generator, and the tanh function in output**
- **Introduce batch normalization**
- **Use LeakyReLU activations for the hidden layers of the discriminator**







Stride [10]

Batch Norm [11]

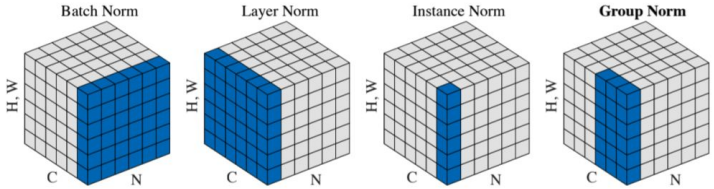


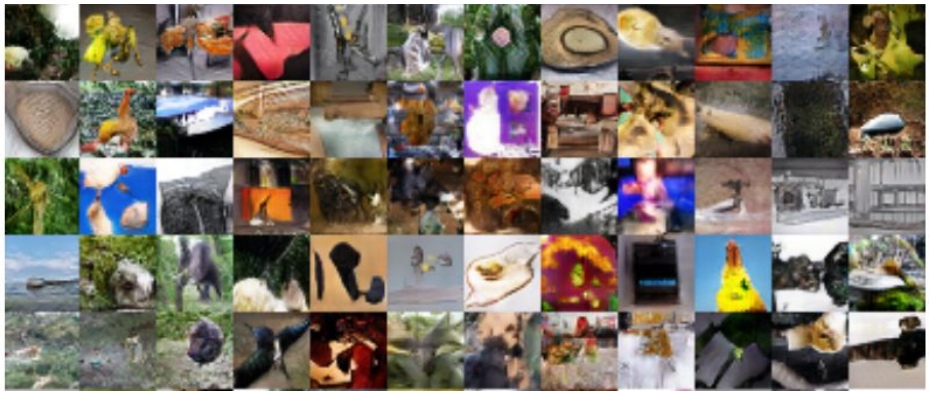
Figure 2. **Normalization methods**. Each subplot shows a feature map tensor, with N as the batch axis, C as the channel axis, and (H, W) as the spatial axes. The pixels in blue are normalized by the same mean and variance, computed by aggregating the values of these pixels.

For full list of references visit: https://bit.ly/3x5DycJ

(Amar Meddahi 2021)

DCGAN (2016)

Some results from the article [10]



Generations of a DCGAN that was trained on the Imagenet-1k dataset

Face generations

For full list of references visit: https://bit.ly/3x5DycJ

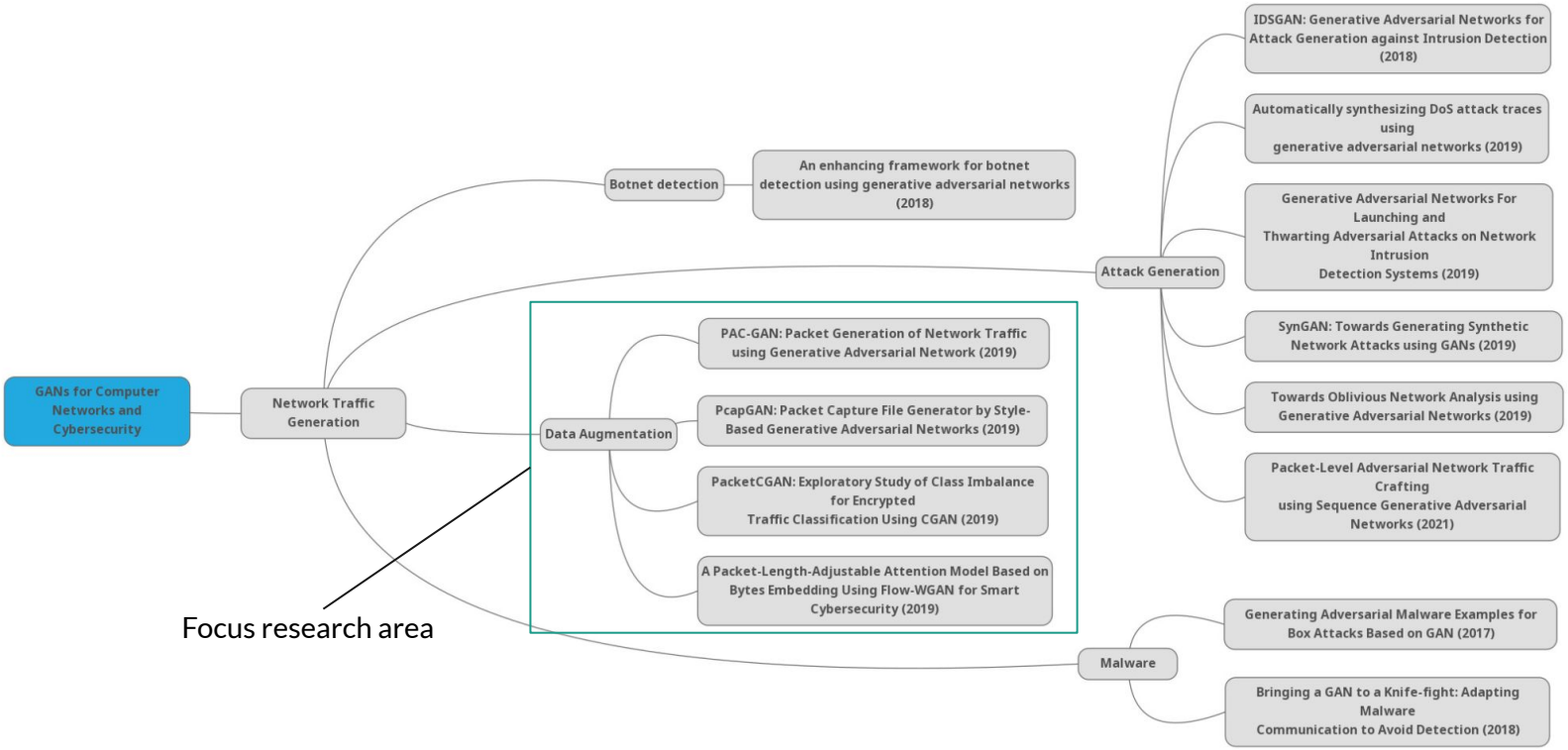
(Amar Meddahi 2021)

GANs for Network Security

For full list of references visit: https://bit.ly/3x5DycJ

(Amar Meddahi 2021)

Literature Review

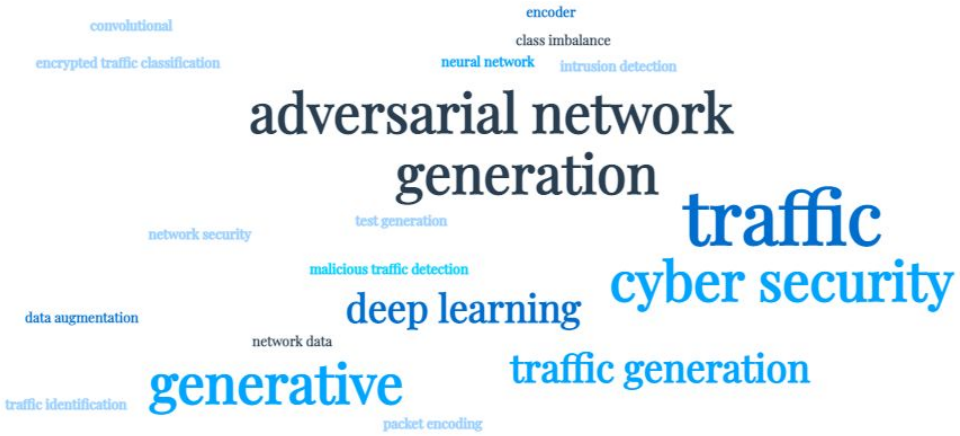


GANs for Computer Networks and CyberSecurity: Tree of Publications

For full list of references visit: https://bit.ly/3x5DycJ

(Amar Meddahi 2021)

Introduction



Word Cloud of the focus research area

Why do we want to generate network traffic with GANs ?

- Generating realistic network traffic is essential for development and testing of techniques in cyber and network security (ex: anomaly or intrusion detection)
- Building a dataset is a time-consuming and costly task
- Synthetic traffic generator are often unrealistic or suffer from a data generation bias
- Commercial test generators can be effective but are costly (ex: IXIA BreakingPoint = $25,000 per year) [12]
- Most of the traditional generators do not allow to analyse packets with tools like Wireshark
- Working on customer databases can pose privacy concerns

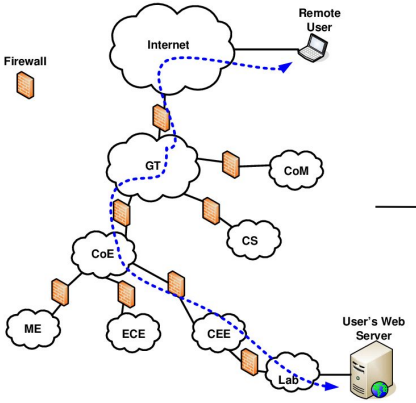
Research challenge

Produce a large corpus of realistic and labelled traffic data using cost-effective means

For full list of references visit: https://bit.ly/3x5DycJ

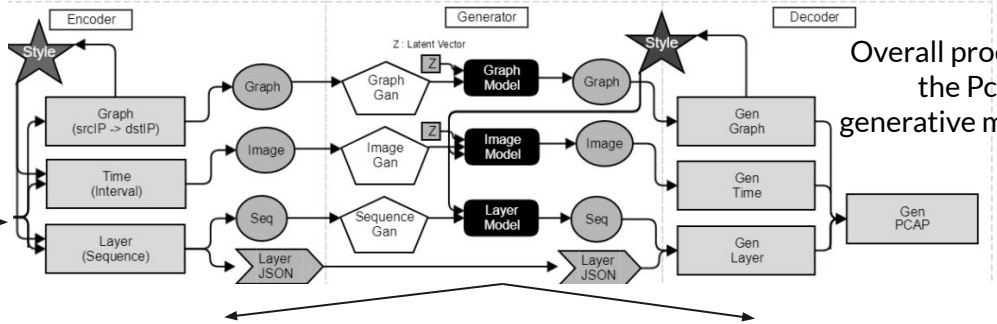
(Amar Meddahi 2021)

Network Traffic Generation : Traffic flow

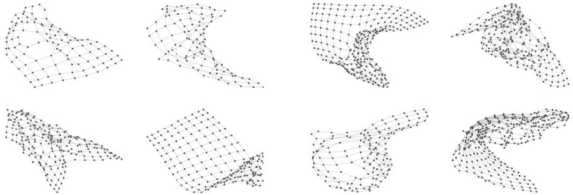


Example of an enterprise network topology [15]

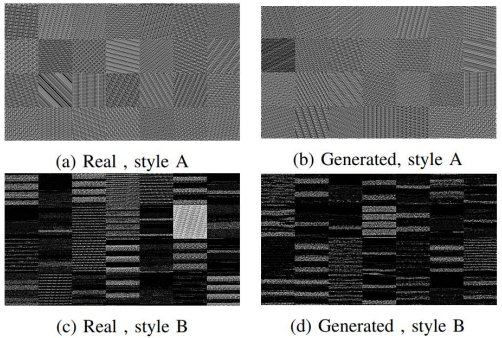
pcap (« packet capture »)



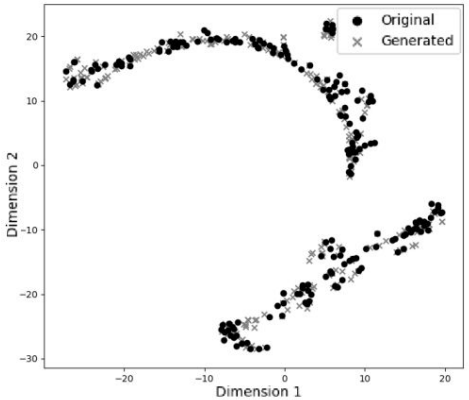
Overall process of the PcapGan generative method [16]



IP graph comparison [16]



Time Image Comparaison [16]

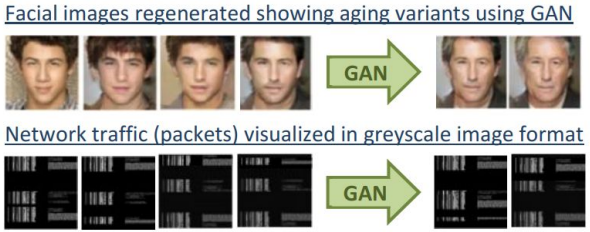


PCA (q=2) to check the validity of the generated data [16]

For full list of references visit: https://bit.ly/3x5DycJ

(Amar Meddahi 2021)

Network Traffic Generation : IP Packet Level



"Question: Can realistic network traffic be generated from GANs? "

Fig. 1. Traffic generation versus image generation using GANs

How CNN GANs can be applied to network traffic generation ? [13]

GAN architecture [13]

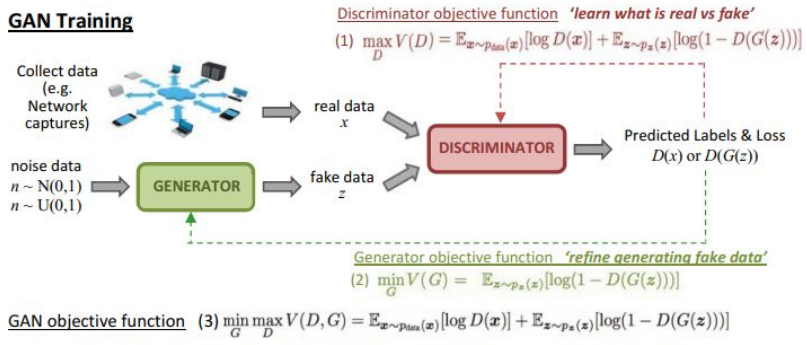


Fig. 2. Objective Functions and Procedural Flows in Training Generative Adversarial Networks

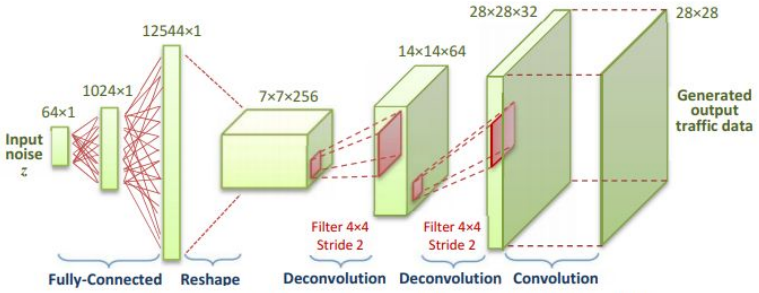


Fig. 3. An Example Generator Model Architecture with Inverse Convolutional Neural Nets

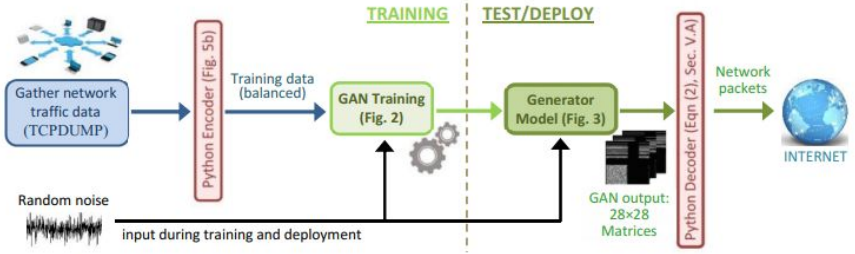


Fig. 4. CNN GAN Traffic Generator Framework with GAN Training and Generator Test/Deployment Phases

Generator Model Architecture with inverse CNN [13]

Training and Generator Test/Deployment [13]

For full list of references visit: https://bit.ly/3x5DycJ

(Amar Meddahi 2021)

Network Traffic Generation : IP Packet Level

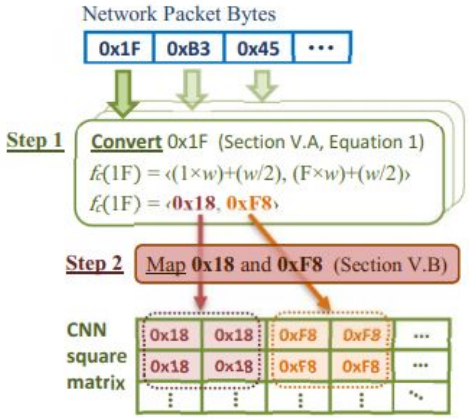


Fig. 5b. Convert & One-to-Multi Map Encoding Process, w=16 d=2

One-to-Multi Map Encoding Process [13]

The encoding scheme consists of two steps:
Step 1: conversion of individual packet byte values for representation by subranges of sequential values
Step 2: duplicating these converted values and mapping them multiple times into the CNN input square matrix (i.e. create 'clusters' of similar converted packet byte values)

Input
Number of times d to duplicate value across each row and col.
Square matrix of size n with $z_{i,j}$ elements at i^{th} row and j^{th} col.
Output
Square matrix with duplicated and mapped byte digit values

1: **Initialize** $i = 0, j = 0$
2: **for** every converted byte digit y_k **do**
3: **for** next sub-matrix $S_{d\times d}$ of d rows and d columns in M **do**
4: Assign $z_{i,j} = y_k$ to all elements of submatrix $S_{d\times d}$
5: Increment i,j to the start of next submatrix $S_{d\times d}$
6: Pad all non-assigned elements $z_{i,j}$ in $M_{n\times n}$ to 0

Packet Byte One-to-Multi Mapping Pseudo Code [13]

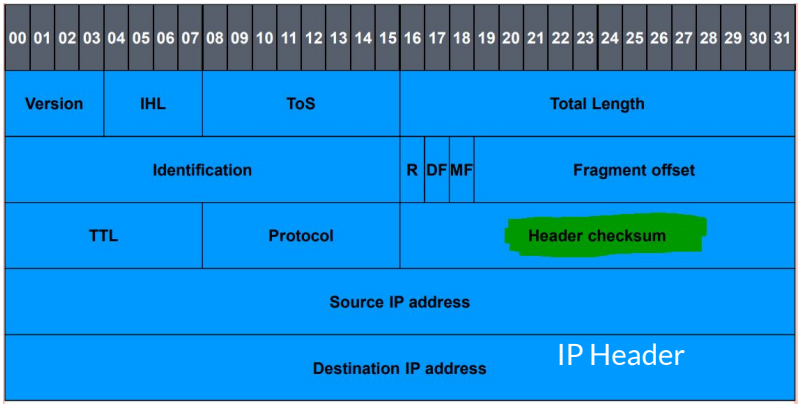
$$f_c(X) = \langle [(x_n \times w) + (w/2)], \cdots, [(x_1 \times w) + (w/2)], [(x_0 \times w) + (w/2)] \rangle$$ Encoding function [13]

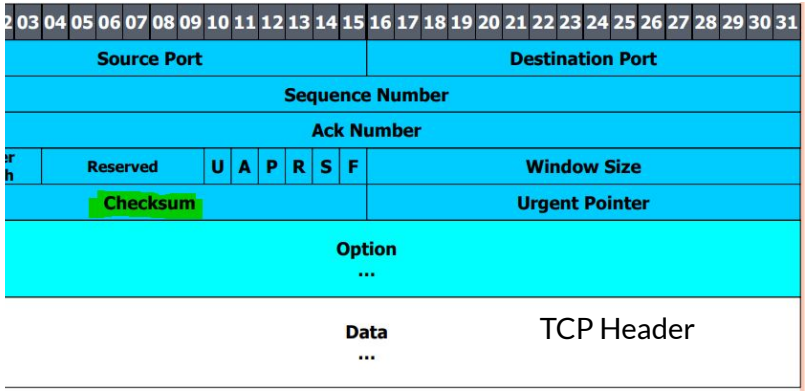
$$f_c^{-1}(Y) = (\lfloor y_n/w \rfloor \times w^n) + \cdots + (\lfloor y_1/w \rfloor \times w^1) + (\lfloor y_0/w \rfloor \times w^0)$$ Decoding function [13]

For full list of references visit: https://bit.ly/3x5DycJ

(Amar Meddahi 2021)

Network Traffic Generation : IP Packet Level





Performance metrics ⇨

Success Rate = number of packets successfully sent / total number of packets generated by PAC-GAN
Byte Error = number of byte field values in a packet that were incorrectly generated (network standards)

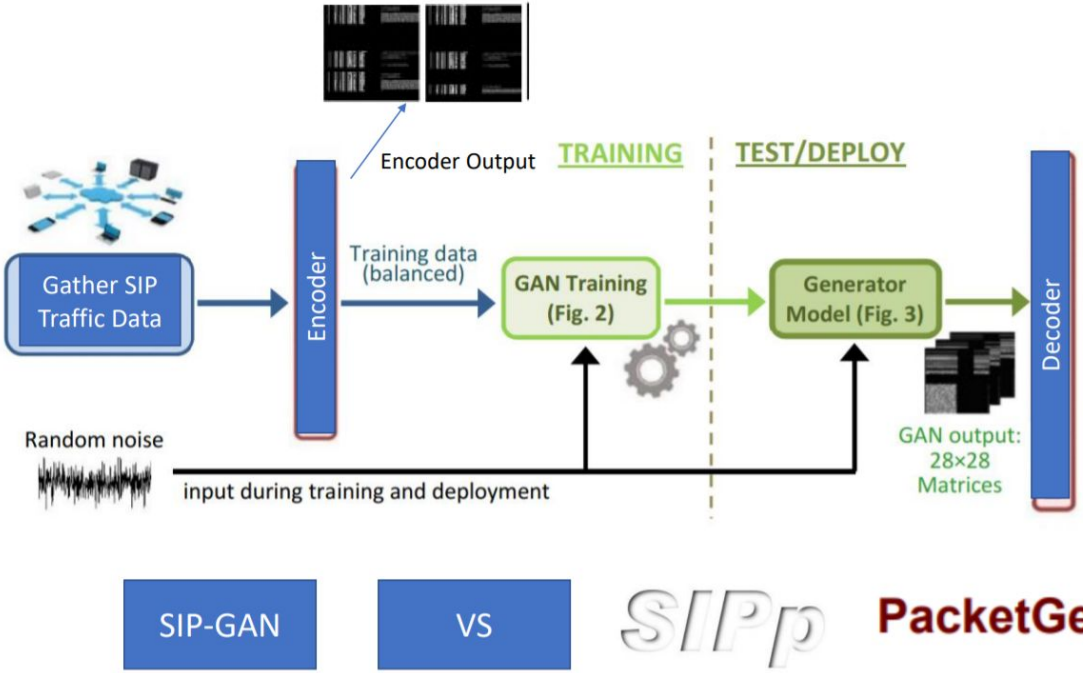
	Ping	DNS	HTTP	Ping/DNS	Ping/HTTP	DNS/HTTP	Ping/DNS/HTTP
Success Rate[1]	76% - 90%	95% - 99%	76% - 79%	75% - 86%	71% - 85%	70% - 88%	66% - 87%
Byte Error[2]	24	0.1	0.4	P:36 D:0.6	P:36 H:1.1	D:0.6 H:0.9	P:12 D:0.2 H:1.9
# Training Steps	12800	19200	19200	20000	22000	24000	28000
Training Time	258mins	313mins	313mins	300mins	369mins	377mins	400mins

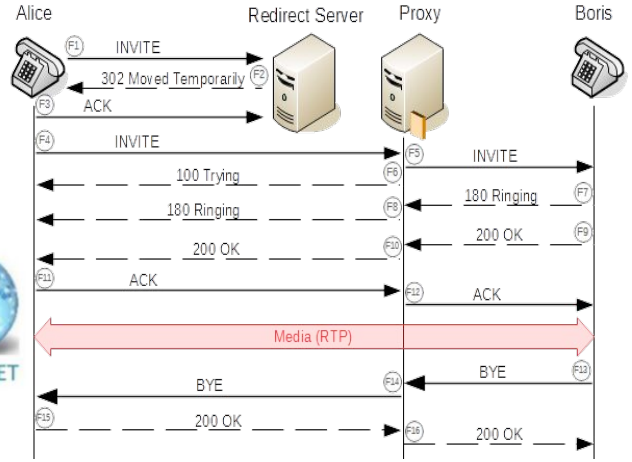
([1] - Best and worst success rates shown. [2] - P, D and H denotes average number of byte field errors per packet for Ping, DNS and HTTP respectively)
Results [13]

For full list of references visit: https://bit.ly/3x5DycJ

(Amar Meddahi 2021)

A new approach for SIP packet generation





Establishment of a session

SIP-GAN

VS

SIPp

PacketGen™

Version		Flow Label
Payload Length	Payload Type	Hop Limit
Source Address		
Destination Address		

32 Bit/s

SIP Header

For full list of references visit: https://bit.ly/3x5DycJ

(Amar Meddahi 2021)

Q&A

For full list of references visit: https://bit.ly/3x5DycJ

(Amar Meddahi 2021)