Generative Adversarial Networks for Failure Prediction (2019)

Shuai Zheng, Ahmed Farahat, and Chetan Gupta

Presented by Kristoffer Gjerde

Norwegian University of Science and Technology

Content

- Motivation
- Problem Description
- Related Work
- Methods
- Experiments and Results
- Conclusion

Motivation

- Prognostics and Health Management (PHM)
    - Failure mechanisms to system lifecycle management
    - Predict failures in the equipment
- Complex physical models → Machine learning
- Three obvious problems
    - Highly imbalanced data
    - Expensive to collect
    - Highly complex or random patterns
- Motivation?
    - Save cost & make the world a safer place
    - Refrigerator example from 2017

Problem Description

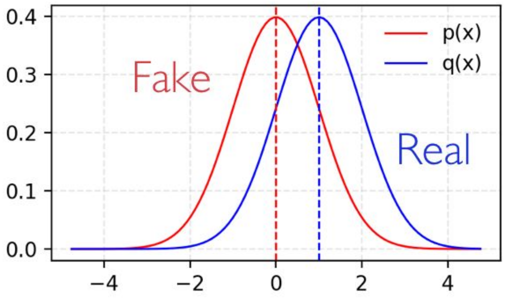
- Failure prediction approaches
  - Model-based
  - Data-driven
- Biggest challenge
  - Failure examples in data set
- How do we solve this?
  - Artificially generate failure data
  - Machine learning
  - Generative Adversarial Networks (GANs)

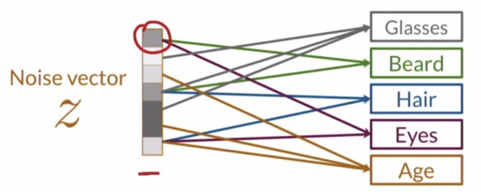
Related Work

- Handling imbalanced data
  - Resampling (over-/undersampling)
    - SMOTE (Synthetic Minority Oversampling Technique)
      - closest neighbors
    - ADASYN (Adaptive Synthetic)
      - “harder-to-learn” neighborhoods
  - Cost-sensitive learning
    - Misclassification costs

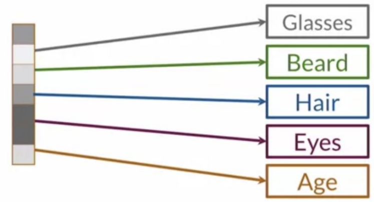
Generative Adversarial Networks

- Minimax two-player game
  - **Generator** - captures data distribution
  - **Discriminator** - probability for real/fake
- Goal:
  - Generator learns the real data distribution
- InfoGAN (2016)
- Conditional GAN (2014)



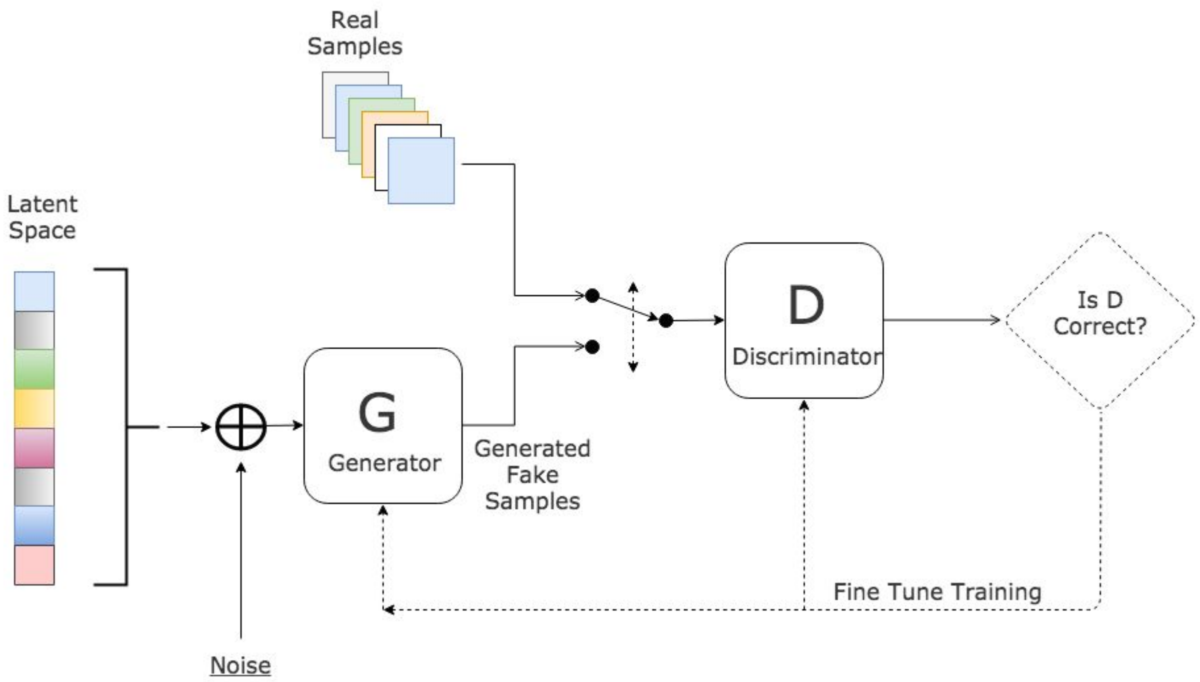


vs.



NTNU

Vanilla GAN



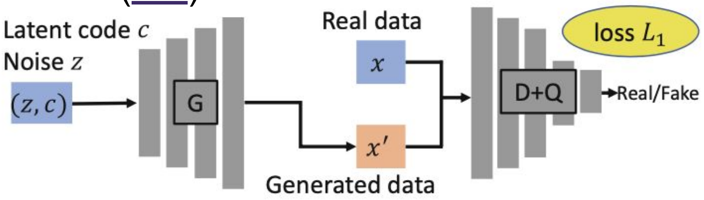
Real Samples
Latent Space
G
Generator
Generated Fake Samples
D
Discriminator
Is D Correct?
Fine Tune Training
Noise

Original GAN (link) $$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\log(1-D(G(\mathbf{z})))]. \quad (1)$$

InfoGAN vs. Conditional GAN

mutual information between c and G(z,c)

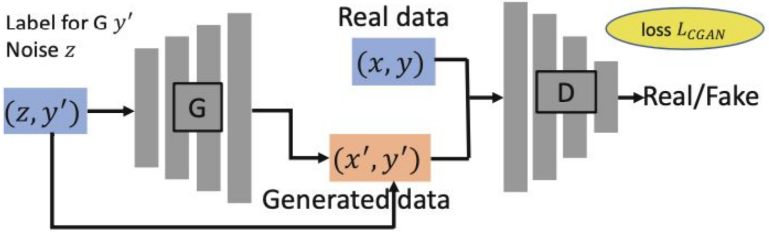
InfoGAN (link)



$$\min_G \max_D V_I(D,G) = V(D,G) - \boxed{\lambda I(c; G(z,c))}$$

$$\min_{G,Q} \max_D V_{\mathrm{InfoGAN}}(D,G,Q) = V(D,G) \boxed{- \lambda L_I(G,Q)}$$

CGAN (link)

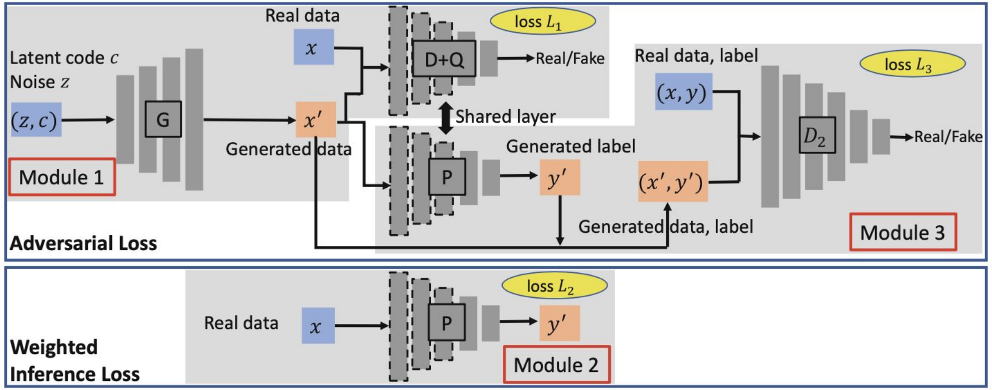


$$\min_G \max_D L_{CGAN}(D,G) = \mathbb{E}_{(\mathbf{x},y)\sim p(\mathbf{x},y)}[\log D(\mathbf{x},y)] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\log(1-D(G(\mathbf{z},y'),y'))]. \quad (4)$$

NTNU

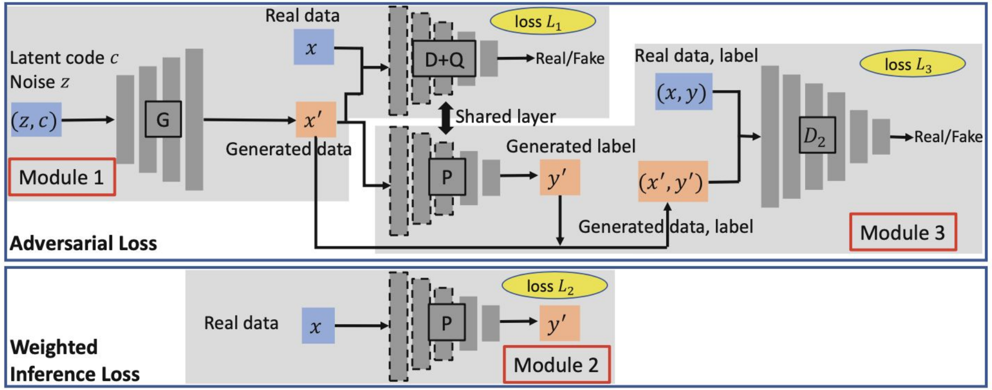
Methods - GAN-FP

- Module 1
  - InfoGAN
  - Generate data = balanced
- Module 2
  - Binary Classifier
    - Weight init from D + Q
  - Imbalanced data
  - Weighted BCE Loss
- Module 3
  - Generated data input
  - P generate labels
  - Enforce generated data-label pair



G: Generator
D + Q: Discriminator
P: Binary Classifier (& Generator in 3)
D2: Discriminator

Methods - GAN-FP



(1) InfoGAN: $$\min_{G,Q} \max_{D} L_1(D, G, Q) = V(D, G) - \lambda_Q L_{mutual}(G, Q).$$

(2) Weighted BCE: $$\min_{P} L_2(P) = \mathbb{E}_{(\mathbf{x},y)\sim p(\mathbf{x},y)}[-wy\log(P(\mathbf{x})) - (1-y)\log(1-P(\mathbf{x}))],$$

$$w = \frac{\text{number of non-failure samples}}{\text{number of failure samples}} > 1$$

(3) Conditional GAN: $$\min_{P} \max_{D_2} L_3(P, D_2) = \mathbb{E}_{(\mathbf{x},y)\sim p(\mathbf{x},y)}[\log D_2(\mathbf{x}, y)] + \mathbb{E}_{\mathbf{x}'\sim p(\mathbf{x}')}[\log(1 - D_2([\mathbf{x}', P(\mathbf{x}')]))].$$

NTNU

Experiments

- Data
  - Air Pressure System (APS)
  - Turbofan Engines (CMAPSS)
- Experimental Setup
  - Fully connected layers
  - ReLU
  - z = 60, c = 4 dim
- Evaluation Criteria
  - AUC (Area Under Curve)
  - Precision, Recall, F1
  - .. with macro-, micro average and failure class only

Table 1. Data sets.

Name	Dimension	Failure sample #	Non-failure sample #	Operating condition #	Fault condition #
APS	170	1,000	59,000	N/A	N/A
CMAPSS FD001	315	2,000	12,031	1	1
CMAPSS FD002	315	5,200	31,399	6	1
CMAPSS FD003	315	2,000	16,210	1	2
CMAPSS FD004	315	4,980	39,835	6	2

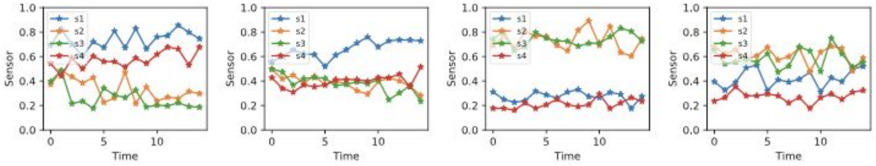
Table 2. Network structures.

Network	APS	CMAPSS
G	64, 64, 170	64, 256, 500, 500, 315
D	170, 64, 1	315, 500, 500, 256, 1
Q	170, 64, 64, 1	315, 500, 500, 256, 64, 1
P	170, 64, 64, 1	315, 500, 500, 256, 64, 1
D_2	171, 64, 1	316, 500, 500, 256, 1

NTNU

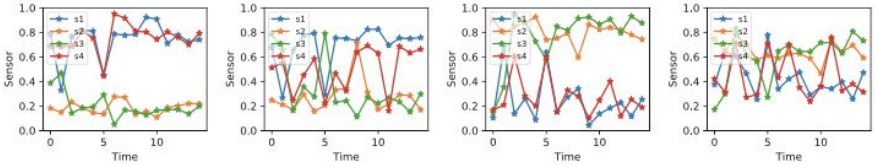
Visualization of Generated Samples

- CMAPSS FD001 data
  - Failure and non-failure
  - Turbofan engine
- Each engine
  - 21 sensors
  - 15 time steps
  - 21 * 15 = 315 (input size of network)
- 4 sensors pr diagram
- 4 real and 4 generated examples
- Captures the major properties



(a) Failure. (b) Failure. (c) Non-failure. (d) Non-failure.

Fig. 4. Real CMAPSS FD001 failure and non-failure samples.



(a) Failure. (b) Failure. (c) Non-failure. (d) Non-failure.

Fig. 5. Generated CMAPSS FD001 failure and non-failure samples.

Results

Classical Methods

Table 3. APS result.

		AUC	Macro			Micro			Failure		
			Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
DNN	Undersampling	0.5751	0.7393	0.8118	0.7705	0.9827	0.9827	0.9827	0.4847	0.6350	0.5498
	Weighted loss	0.6131	0.8027	0.8042	0.8034	0.9871	0.9871	0.9871	0.6119	0.6150	0.6135
	SMOTE	0.7077	0.8434	0.8350	0.8391	0.9896	0.9896	0.9896	0.6923	0.6750	0.6835
	ADASYN	0.6971	0.8040	0.8561	0.8279	0.9878	0.9878	0.9878	0.6128	0.7200	0.6621
SVM	Undersampling	0.3130	0.6995	0.7706	0.7293	0.9791	0.9791	0.9791	0.4066	0.5550	0.4693
	Weighted loss	0.3004	0.6829	0.7623	0.7151	0.9773	0.9773	0.9773	0.3737	0.5400	0.4417
	SMOTE	0.5673	0.7432	0.8169	0.7749	0.9830	0.9830	0.9830	0.4924	0.6450	0.5584
	ADASYN	0.5188	0.7225	0.8158	0.7606	0.9810	0.9810	0.9810	0.4510	0.6450	0.5309
RF	Undersampling	0.4274	0.6449	0.8813	0.7052	0.9647	0.9647	0.9647	0.2934	0.7950	0.4286
	Weighted loss	0.3750	0.6838	0.7333	0.7054	0.9781	0.9781	0.9781	0.3765	0.4800	0.4220
	SMOTE	0.4137	0.6602	0.7414	0.6919	0.9747	0.9747	0.9747	0.3289	0.5000	0.3968
	ADASYN	0.3387	0.6302	0.8360	0.6832	0.9626	0.9626	0.9626	0.2655	0.7050	0.3858
DT	Undersampling	0.5614	0.5928	**0.9330**	0.6376	0.9311	0.9311	0.9311	0.1868	**0.9350**	0.3114
	Weighted loss	0.6310	0.8194	0.8022	0.8106	0.9879	0.9879	0.9879	0.6455	0.6100	0.6272
	SMOTE	0.6471	0.7751	0.8625	0.8125	0.9858	0.9858	0.9858	0.5547	0.7350	0.6323
	ADASYN	0.6094	0.7567	0.8420	0.7930	0.9842	0.9842	0.9842	0.5187	0.6950	0.5940
InfoGAN AUG		0.7343	0.8335	0.8744	0.8527	0.9898	0.9898	0.9898	0.6711	0.7550	0.7106
GAN-FP		**0.8085**	**0.8662**	0.8955	**0.8803**	**0.9918**	**0.9918**	**0.9918**	**0.7358**	0.7959	**0.7647**

NTNU

Conclusion

- GAN
- Imbalanced dataset
- Failure prediction
- Experiment on industrial data
- Improved modeling performance
- Big potential - the original motivation
  - Save money
  - Save lives