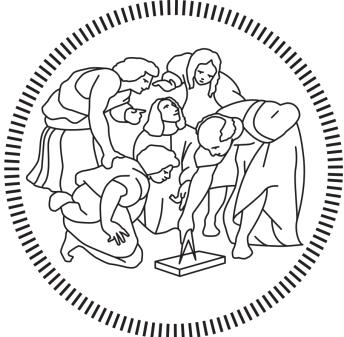


POLITECNICO
MILANO 1863

DEPARTMENT OF
ELECTRONICS, INFORMATICS AND BIOENGINEERING
M.Sc. COURSE OF COMPUTER SCIENCE AND ENGINEERING



**Adversarially Learned Anomaly Detection
Using Generative Adversarial Networks**

Supervisor:

Doc. Giacomo BORACCHI — Politecnico di Milano

Master of Science Thesis of:
Şemsi Yiğit ÖZGÜMÜŞ
Matriculation ID 893558

July 2019

Acknowledgements

 Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Contents

Contents	iv
List of Figures	vii
List of Tables	vii
1 Introduction	5
2 State of the Art	7
2.1 Anomaly Detection	7
2.2 Architectural Components	11
2.2.1 Generative Adversarial Networks	11
2.2.2 Autoencoder Networks	17
2.3 GAN Based Anomaly Detection Methods	20
2.3.1 AnoGAN	21
2.3.2 BiGAN and ALI	23
2.3.3 ALAD	30
2.3.3.1 Spectral Normalization	30
2.3.3.2 Conditional Entropy (ALICE Framework) .	31
2.3.3.3 ALAD Architecture	33
2.3.4 GANomaly & Skip-GANomaly	35
3 Latest Developments	41
3.1 F-AnoGAN Framework	41
3.2 Energy Based Generative Adversarial Networks	45
4 Architectural Improvements	49
4.1 SEM Images Dataset	49
4.2 Analysis of Aforementioned Approaches	49
4.3 Encoded Energy Based Generative Adversarial Network . . .	49
4.4 Sequentially Encoded Energy Based Generative Adversarial Network	49

Contents

5 Experimental Results	51
5.1 Experiment Settings	51
5.1.1 Figure of Merit	51
5.2 GAN Based Model Analysis	51
5.3 AR Based Model Analysis	53
5.4 Improved Model Analysis	53
5.5 Discussion of Experiment Results	53
6 Conclusion and Future Work	55
A Model Implementation Details	57
A.1 AnoGAN Implementation	57
A.2 BiGAN Implementation	58
A.3 ALAD Implementation	58
A.4 GANomaly Implementation	58
A.5 Skip-GANomaly Implementation	61
A.6 EBGAN Implementation	61
A.7 SENCEEBGAN Implementation	61
Bibliography	63

List of Figures

21	GAN architecture overview	12
22	Simple Autoencoder architecture	18
23	Stacked Autoencoder with multiple layers	18
24	ALI architecture overview	23
25	BiGAN architecture overview	24
26	ALAD architecture overview	33
27	GANomaly architecture overview	36
28	Skip GANomaly architecture overview [3]	38
31	F-AnoGAN Framework Overview [41]	42
32	F-AnoGAN Training Strategies [41]	43
33	Anomaly Score computations for both training methods [41] . .	44
34	EBGAN Model Structure [49]	46

List of Tables

51	Ablation study for AnoGAN to test the effect of various training improvements for stabilization.	51
52	Ablation study for BiGAN to test the effect of various training improvements for stabilization.	52
53	Ablation study for ALAD to test the effect of various training improvements for stabilization.	52
54	Ablation study for GANomaly to test the effect of various training improvements for stabilization.	53

LIST OF TABLES

55	Ablation study for Skip-GANomaly to test the effect of various training improvements for stabilization.	53
A1	Architecture and Hyperparameters of AnoGAN Model	57
A2	Architecture and Hyperparameters of BiGAN Model	58
A3	Architecture and Hyperparameters of ALAD Model	59
A4	Architecture and Hyperparameters of GANomaly Model	60
A5	Architecture and Hyperparameters of Skip-GANomaly Model . .	61
A6	Architecture and Hyperparameters of EBGAN Model	62
A7	Architecture and Hyperparameters of SENCEEBGAN Model . .	62

Abstract

 Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Sommario

 Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

1

CHAPTER

Introduction

Introduction will be written

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fer-

1. INTRODUCTION

mentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

This document is organized as follows:

Chapter 2 outlines the state of the art methods that is used with this type of dataset. Chapter 3 explains other anomaly detection methods that benefit from the generative adversarial networks. Chapter 4 presents the anomaly detection problem thoroughly and reconstruction based approaches and the proposed approach for the problem. Experimental results are described in Chapter 5 and the conclusion and the future work is stated in chapter 6.

CHAPTER 2

State of the Art

This chapter is dedicated to the exploration of state of the art approaches in the fields relevant to the thesis. The organization of the chapter is the following: Section 2.1 will define what anomaly detection is and explain different approaches towards its solution. Section 2.2 will explain what SEM image is, how it's acquired, define the anomaly detection problem in that context and explain state of the art methods that are used specific to this problem. Section 2.3 will give necessary background information about Generative Adversarial Networks (GAN) and Autoencoder Networks. First the principles behind how GANs work will be presented then the autoencoder network architecture will be explained. This section is particularly significant to relate how different architectures that uses these networks function. In section 2.4, variety of architectures that is developed for anomaly detection will be studied as state of the art. How they are implemented and how they work in practise will be described.

2.1 Anomaly Detection

Using the terminology of machine learning, anomaly detection can be defined as the task of classifying test data that differs in some respect from the data for the model that is availabla during training. [35] This definition correctly describes the task on the abstract level but does not clarify what anomaly actually is. Depending on the the application domain, anomalies can be referred to as outliers, novelties, discordant observations, exceptions, aberrations, surprises, peculiarities and contaminants.[10] Sometimes these can be used interchangeably in the context. These distinctions in the defi-

2. STATE OF THE ART

nition are primarily relies not only on the features of the data such as its dimensional complexity, its continuity and its format, but also on what is considered an anomaly for that specific domain. This makes it harder to conceptualize the problem of anomaly detection with a general definition and a solution approach that can encompass all different domains.

In the rest of this section, I will explain different use cases for anomaly detection to show its impact on a broad spectrum of problems then the primary approach types to solve anomaly detection problem will be introduced.

The categorization threshold to define different application domains is endless but the use of anomaly detection can be categorized down to these six general fields. [35] These are:

- IT Sector
- Healthcare informatics, medical diagnostics and monitoring
- Industrial Monitoring/ maintainence and damage detection
- Image processing/ Video Surveillance
- Sensor Networks

IT Sector

Anomaly detection methods have a huge impact on the IT security systems, in which the objectives include network intrusion detection and fraud detection. [35] With the increased usage of computerized systems and the inclusion of the internet based services to our daily lives, maintainability and the availability of these systems became much more important. For systems like these, anomaly is considered as a priority because it indicates significant but rare events and can prompt to critical actions to be taken in a wide range of application domain. [1]. For example a malicious process in a server may show increased use of resource from time to time that may be spotted as a type of anomaly in the system resource use patterns or the unusual amount of traffic on the network may indicate an intruder attack. [15] [22] Another aspect of the increased use of internet is its impact on the e-commerce field and dramatically growing online payment systems. This also includes a migration to internet banking in favor of the convenience of technology. As a result, insurance and credit card frauds, scams towards phone and mobile data usage, and phishing attacks targeting financial transactions become more and more severe. [16] In this context, behaviour of the consumer becomes the data. Transaction history, time spent between consecutive transactions and the geographic data can be used to construct a model to detect anomalies in the purchase patterns.

Healthcare

Healthcare and medical diagnostics is another domain for anomaly detection. Detecting anomalies in the progression of the disease symptoms is a crucial factor for treatment monitoring. [42] To aid diagnostics, interpretation of the patient data is also an important task for anomaly detection. Ability to differentiate between the instrumentation or recording errors and the actual clinically relevant change that may indicate a symptom should also be present in these anomaly detection systems. [35]. Patient data may include tabular information like the age, weight, blood test results, heart rate or respiratory rate [11][30] or medical image data [42].

Industrial Monitoring

Novelty detection became a huge part of the industrial automation systems from the quality control of the manufacturing progress to the performance maintainance of the equipments. Materials that have a complicated and delicate manufacturing process like cpu wafers [23] and nanofibrous materials [33] may easily develop defects in the manufacturing pipeline. Therefore using anomaly detection is a viable option to reduce the wasted material and optimize the cost of producing the target product. Same principle also applies to the equipment used in these pipelines. Equipments and assets have the risk of deteriorating over time due to the usage and wear [35]. For example heavy extraction machines that is used in the petroleum industry like turbomachines [31] needs constant supervision for damage prevention.

Image Processing

Image processing methods has been used with anomaly detection methods extensively to detect novelty subjects or anomalies in the image or in a video stream. Detection in video data stream data is an important task for the surveillance systems that is used with the purpose of traffic control, security systems and search and rescue. [4]

Sensor Networks

With the improvements of the smart devices' involvement in our daily lives and the recent but promising developments in the field of internet of things, the definition of acquiring and processing data is evolved. Various types of sensors, especially wireless, are heavily used to collect huge amount of data to feed various systems such as surveillance, environmental monitoring, and disaster management. [45] These sensors usually used to create a network distributed over a large area with one or depending on the scale multiple sink nodes responsible for gathering information from the other

2. STATE OF THE ART

sensors. In these distributed systems, sensors are used to compute or predict information depending on the problem. However the accuracy of these computations or predictions depends primarily on the quality of the sensor measurements and with this scale the instrumentation error is unavoidable. That's why anomaly detection tasks usually deployed on these systems to observe sensory faults and prevent malicious attacks. [35] [7]

Variety of approaches that belong to different sub domains of the machine learning, statistics , bayesian and information theory are used to solve anomaly detection problem for these aforementioned fields. All these different approaches in general tries to build a model or a representation of the training data which contains no anomalous samples (or very few) and assigns them an anomaly score or a label depending on the problem. At the inference stage, deviations from the constructed normality definition are detected using a predetermined anomaly threshold. These different approaches can be categorized into 5 main titles. These are:

- Probabilistic Anomaly Detection Methods
- Distance Based Anomaly Detection Methods
- Domain Based Anomaly Detection Methods
- Information Theoretic Anomaly Detection Methods
- Reconstruction Based Anomaly Detection Methods

Probabilistic approaches aim to estimate the generative probability density function of the data with the assumption that the input data is generated from an unknown probability distribution D , and learning this distribution gives us an idea about the representation of normality for the input data, training set. Anomaly (or novelty in some approaches) threshold then can be determined which has certain probabilistic interpretation and can be used to detect the anomaly on the inference stage. [35] Distance based methods are another type of modelling technique used to model the training data with a probability density function equivalent process. Including the K-Nearest neighbour based and clustering based methods, distance based methods rely on distance metrics between the data points to create a representation for normality. For example considering the K-NN aproach, data points in the training dataset are identified to have close neighbour points using the distance metrics while if a sample has a greater distance to the nearest normal point with a value above the determined anomaly threshold, that sample is identified as an anomaly. Domain based methods works by creating a boundry based on the structure of the training dataset. [35] These methods are usually insensitive to different sampling or the density properties because instead of learning the density of the data, the learn

the boundary information to separate the target class (anomalies) from the normal data. Support Vector Machines (SVM) are the popular example for this approach. Information theoretic anomaly detection methods computes the total information content that resides in the training dataset. The main assumption is that anomalous samples present in the data significantly modifies the information content of the assumed normal dataset. Metrics for this approach are calculated using the entirety of the dataset and the subset that triggers the biggest discrepancy is identified as the anomaly.

Finally reconstruction based approaches which also is the focus of this thesis, model the underlying representation of the data and reconstructs it. Using the original input and the reconstructed output, these models define a reconstruction error tailored to the nature of the problem and the type of data and measures this error with respect to certain anomaly threshold. Since the modelled underlying data has the features specific to the training dataset, when the anomalous sample is feeded through this pipeline, its representation and consequently its reconstruction will be made poorly hence giving a greater reconstruction error. In the later sections, state of the art generative adversarial network based frameworks that uses this approach will be introduced. How the reconstruction loss is computed and anomaly score is formed will be explained in each framework.

2.2 Architectural Components

This section will explain the two main network types used in the state of the art anomaly detection frameworks in the following sections. Generative adversarial networks section will describe how the framework is formed, the intuition behind its adversarial training scheme. Autoencoder networks section will give an introductory information to the subject and introduce different kinds of approaches that can be integrated into both the construction of the network and its training methodology.

2.2.1 Generative Adversarial Networks

One of the main objectives of the deep learning is to be able to create complex and high dimensional models that can capture the relations amongst the fundamental types of data that we perceive and be exposed to in our daily lives. This data can vary, it can be a visual information like an image, symbols in natural language or audio waveforms containing speech. Then we use these models to make machines make sense of the world we live in and exhibit what we call an intelligence. [8]

2. STATE OF THE ART

In the terms of statistical classification, also in machine learning (that consequentially includes deep learning) the main two approaches are called discriminative and generative modelling. Discriminative models maps the rich, complex sensory data to a class label, target so to speak to perform the desired task. They try to learn the conditional probability distribution of the input data using the features of the data. The success behind this approach is the result of the improvements in the training, learning and generalization of these models. Backpropagation [46], dropout [44], and rectified linear units[17] are some of the examples of concepts that contribute to this success.

Generative models in that regard did not show a considerable promise as opposed to discriminative models mainly because of the type of problem they try to solve and computational difficulties regards to that solution. The generative models try to estimate the joint probability distribution of the high dimensional sensory input data to be able to generate data similar to input and its label with some variations which can be thought as imaginative. The main difficulty resides in the probabilistic computations that are hard to control in the maximum likelihood estimation. [19] [39]

Generative Adversarial Networks (GAN) are considered as one of the most significant breakthroughs in deep learning and generative modelling in the recent years. It is a framework that consists of two separate modules. One of them is labelled as generator and the other is the discriminator. You can observe the example architecture in the figure 21.

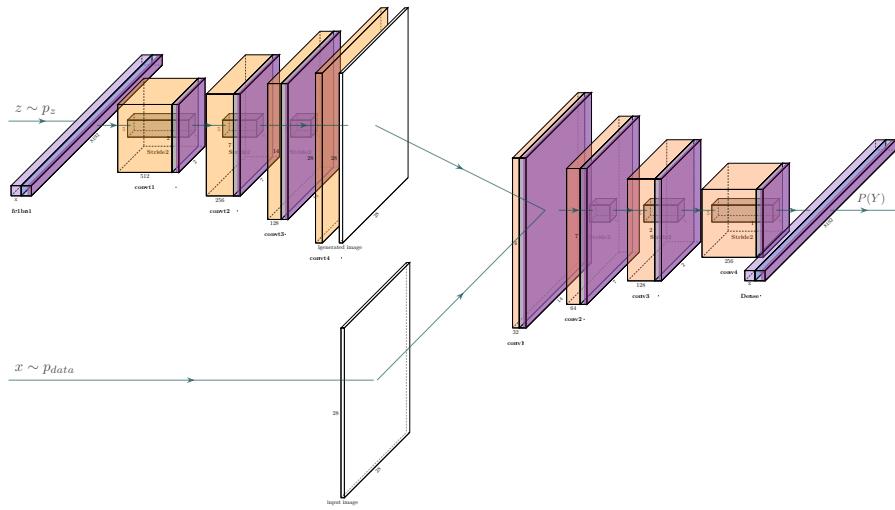


Figure 21 – GAN architecture overview

2.2. Architectural Components

From this point on, I will use "image" to specify the type of the input and the target data because GAN frameworks are mainly designed to work with high dimensional spatial data like images so it will be consistent when explaining the theorem and also the central focus of the thesis is on the spatial data.

The power of this framework comes from the adversarial nature of the training process. Simply put, the goal of the generator network is to create images similar to the input data by capturing the data distribution using a probabilistic model. On the other hand discriminator tries to detect fake (generated) images that is generated by generator from the real input images. We can represent this intuition with the following two player minimax game with the value function $V(G, D)$ in equation 2.3. [19]

$$p_{\text{data}}(x) : x \in \Omega_X \quad (\text{Distribution of the data}) \quad (2.1)$$

$$p_z(z) : z \in \Omega_Z \quad (\text{Distribution of the noise}) \quad (2.2)$$

$$V(G, D) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))] \quad (2.3)$$

The training objective criterion for the discriminator is to maximize the expected log-likelihood of classifying the input data as real. p_{data} represents the distribution of the input. Since the generated data should be classified as fake by the discriminator, the output of the second term is flipped. The Generator's purpose is to "fool the discriminator". Hence its objective criterion is to minimize the expected log likelihood of discriminator classifying generated image sample as fake. We can interpret this equation as two separate objective functions with different training criterions :

$$\min_G V(G, D) = \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))] \quad (2.4)$$

$$\max_D V(G, D) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))] \quad (2.5)$$

While this equation describes the relation between the two adversaries, it does not provide a definite explanation how this training objective finds equilibrium and whether that equilibrium also provides us with the optimal generator or not. For this analysis we will use bottom up approach and prove the theorem mentioned in the original GAN paper. [19]

Theorem 2.2.1. *The global minimum of the virtual training criterion $C(G)$ is achieved iff $p_g = p_{\text{data}}$. At that point, $C(G)$ achieves the value $-\log(4)$.*

We start by defining the term expected value of a function.

2. STATE OF THE ART

Definition 2.2.1. $E(f(x))$ of some function $f(x)$ with respect to certain probability distribution $p(x)$ is called the expected value of a function and it is expressed like

$$E_{x \sim p_x} = \int p(x)f(x)dx \quad (2.6)$$

So the minimax game equation can be expressed like this:

$$V(G, D) = \int_x p_{\text{data}}(\mathbf{x}) \log(D(\mathbf{x})) dx + \int_z p_z(\mathbf{z}) \log(1 - D(G(\mathbf{z}))) dz \quad (2.7)$$

For generator to sample an image good enough to fool the discriminator, framework needs to learn the implicitly created sample distribution p_g by the generator G . Framework defines a prior noise distribution $z \sim p_z$, isotropic gaussian distribution with zero mean and unit variance to learn the distribution p_g (Different distributions are also considered). Next we use the LOTUS theorem.

Theorem 2.2.2. *Law of the unconscious statistician (LOTUS) theorem is used to compute the expected value of a function $g(x)$ of a random variable X when one knows the probability distribution of X but does not know the probability distribution of $g(x)$. [37]*

The expected value of a function $g(x)$ then can be expressed like:

$$E[g(X)] = \sum_x g(x)f_X(x) \quad (2.8)$$

$f_X(x)$ being the probability distribution of the random variable X .

Using this law we can write the equation $V(G, D)$.

$$z \sim p_z, \quad G(z) = x, \quad x \sim p_g \quad (2.9)$$

$$V(G, D) = \int_x p_{\text{data}}(\mathbf{x}) \log(D(\mathbf{x})) + p_g(\mathbf{x}) \log(1 - D(\mathbf{x})) dx \quad (2.10)$$

For the Discriminator D , the value of the training criterion should be maximized. That basically means the derivative of the function of this expected value expression should be equal to zero. We write the function by reversing the process of the equation 2.6.

$$f(p_{\text{data}}, p_g) = p_{\text{data}}(\mathbf{x}) \log(D(\mathbf{x})) + p_g(\mathbf{x}) \log(1 - D(\mathbf{x})) \quad (2.11)$$

2.2. Architectural Components

$$f' = p_{\text{data}}(\mathbf{x}) \frac{1}{D(\mathbf{x}) \ln(C)} - p_g(\mathbf{x}) \frac{1}{(1 - D(\mathbf{x})) \ln(C)} \quad (2.12)$$

$$f' = 0 \quad (2.13)$$

$$p_{\text{data}}(\mathbf{x})(1 - D(\mathbf{x})) \ln(C) = p_g(\mathbf{x})D(\mathbf{x}) \ln(C) \quad (2.14)$$

$$(p_{\text{data}}(\mathbf{x}) + p_g(\mathbf{x}))D(\mathbf{x}) = p_{\text{data}}(\mathbf{x}) \quad (2.15)$$

$$D_G^*(\mathbf{x}) = \frac{p_{\text{data}}(\mathbf{x})}{(p_{\text{data}}(\mathbf{x}) + p_g(\mathbf{x}))} \quad (2.16)$$

This is the optimal discriminator D with the fixed G . By deriving this value, we show that the maximum value the Discriminator can reach is the result of the equation 2.16. Next, we reformulate our $V(G, D)$ to analyze for the training criterion of the generator G .

$$C(G) = \max_D V(G, D) \quad (2.17)$$

$$= \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log D_G^*(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}} [\log (1 - D_G^*(G(\mathbf{z})))] \quad (2.18)$$

$$= \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log D_G^*(\mathbf{x})] + \mathbb{E}_{\mathbf{x} \sim p_g} [\log (1 - D_G^*(\mathbf{x}))] \quad (2.19)$$

$$= \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} \left[\log \frac{p_{\text{data}}(\mathbf{x})}{P_{\text{data}}(\mathbf{x}) + p_g(\mathbf{x})} \right] + \mathbb{E}_{\mathbf{x} \sim p_g} \left[\log \frac{p_g(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_g(\mathbf{x})} \right] \quad (2.20)$$

In the theorem the equality $p_{\text{data}} = p_g$ is given. By applying this equality we get $D_G^*(\mathbf{x}) = \frac{1}{2}$. By putting this to our equation we get

$$C(G) = \log\left(\frac{1}{2}\right) + \log\left(\frac{1}{2}\right) = -\log(4)$$

. To verify that this value is indeed the global point for the equation we apply the following subtraction:

$$\begin{aligned} C(G) - (-\log(4)) &= \int p_{\text{data}} \left[\log \frac{p_{\text{data}}}{p_{\text{data}} + p_g} + \log 2 \right] + \\ &\quad \int p_g \left[\log \frac{p_g}{p_{\text{data}} + p_g} + \log 2 \right] \\ C(G) - (-\log(4)) &= \int p_{\text{data}} \left[\log \frac{2p_{\text{data}}}{p_{\text{data}} + p_g} \right] + \int p_g \left[\log \frac{2p_g}{p_{\text{data}} + p_g} \right] \end{aligned}$$

2. STATE OF THE ART

$$C(G) - (-\log(4)) = \int p_{\text{data}} \left[\log \frac{p_{\text{data}}}{(p_{\text{data}} + p_g)/2} \right] + \int p_g \left[\log \frac{p_g}{(p_{\text{data}} + p_g)/2} \right] \quad (2.21)$$

To interpret this result we introduce 2 divergence definitions.

Definition 2.2.2. The Kullback-Leibler divergence is the measure of how one probability distribution is different from the other one. Its general definition is

$$D_{KL}(P\|Q) = \int p(x) \log \frac{p(x)}{q(x)} \quad (2.22)$$

More generally if P and Q are probability measures¹ over a dataset X , and measure P is absolutely continuous with respect to Q , then the Kullback-Leibler divergence can also be defined as in equation 2.23

$$D_{KL}(P\|Q) = \int_X \log \left(\frac{dP}{dQ} \right) dP \quad (2.23)$$

where $\frac{dP}{dQ}(f_{PQ})$ is the Radon-Nikodym derivative [9]. The defining property of radon-nikodym property that ties to the probability measure is that

$$P(R) = \int_R f_{PQ} dQ \quad (2.24)$$

The RN derivative $f_{PQ} : \Omega \mapsto \mathcal{R}_{\geq 0}$ is defined for any measures P and Q on a space Ω such that P is absolutely continuous with respect to Q . In other words, for any $R \subseteq \Omega : P(R) > 0 \implies Q(R) > 0$. [9]

Definition 2.2.3. Jensen-Shannon divergence is a measure of similarity between different probability distributions. Its major difference from the KL divergence is that it is always symmetric and it always has a non-negative value. The General definition is provided in equation 2.25.

$$JSD(P\|Q) = \frac{1}{2} D_{KL}(P\|M) + \frac{1}{2} D_{KL}(Q\|M) \quad (2.25)$$

¹a probability measure is a real-valued function defined on a set of events in a probability space that satisfies measure properties such as *countable additivity*[38]

Using this divergence measures we can rewrite our equation as this:

$$C(G) = -\log(4) + KL\left(p_{\text{data}} \parallel \frac{p_{\text{data}} + p_g}{2}\right) + KL\left(p_g \parallel \frac{p_{\text{data}} + p_g}{2}\right) \quad (2.26)$$

$$C(G) = -\log(4) + 2 \cdot JSD(p_{\text{data}} \parallel p_g) \quad (2.27)$$

Considering Jensen-Shannon divergence is always non-negative and zero only when the distributions are identical, we can conclude the proof by stating that the global minimum point of $C^*(G) = -\log(4)$ is only achieved when $p_{\text{data}} = p_g$.

Intuition behind the adversarial training of this framework is important because all the subsequent approaches that will be discussed in the thesis that use GAN's fundamentally benefit from this framework to train their networks. As we will see in the BiGAN2.3.2 and similar models, addition of Encoder Network changes the overall minimax game played by the adversary networks but underlying logic still depends on the value function of the GANs.

2.2.2 Autoencoder Networks

Autoencoder network is a type of neural network that is trained to copy its input data to output. [18] In its simplest form, it can be defined as a feedforward, non- recurrent network similar to multilayer perceptrons like in figure 22.

Internally, it consists of a hidden layer that describes the representation (code) learned from the input data. The connections from the hidden layer to the output layer transforms the hidden representation into the output data with the same dimensionality as the input. Rather than trying to reconstruct the given data, autoencoders are mainly used to learn the representation about the data to extract meaningful features and to reduce the dimensionality of the data while keeping the relevant information as an alternative to principal component analysis.

With the increased complexity of the data present, this architecture can be extended to capture more complex feature representations by stacking similar but varying encoder and decoder layers on top of each other. Generally first layers learn the more basic features of the data and as the layers are stacked, more concentrated and complex features can be learnt in the code section of the autoencoder, the end of the encoder module. These

2. STATE OF THE ART

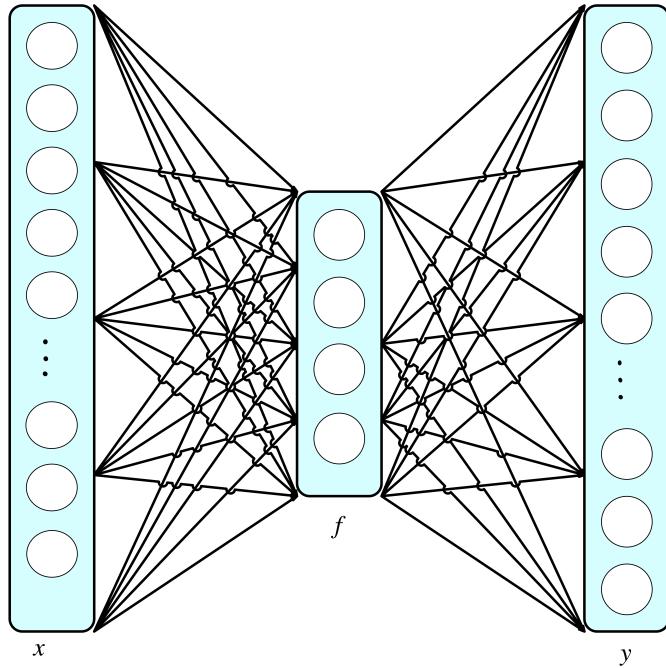


Figure 22 – Simple Autoencoder architecture

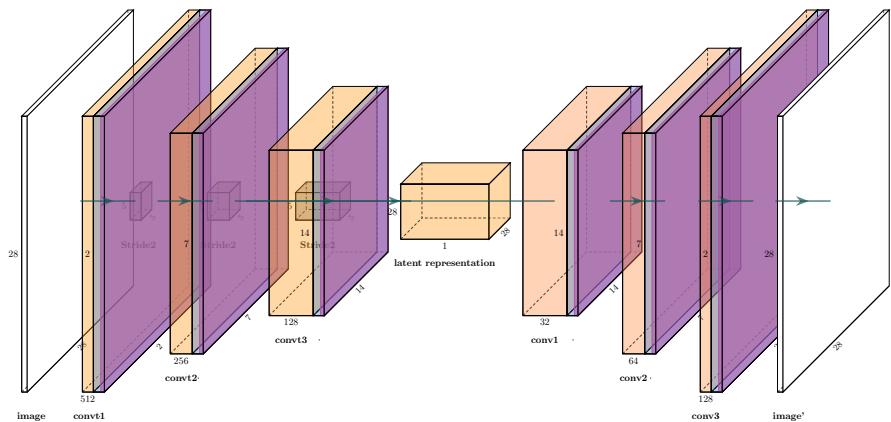


Figure 23 – Stacked Autoencoder with multiple layers

architectures are mainly referred to as deep autoencoders because of the increased level of hidden layers present in the model depicted in figure 23, just like in deep learning.

Learning rationale of the autoencoders can be visualized with a pair of weighted layer computation functions like in the multilayer perceptrons. Assume that we have two transition functions that defines the relationship between the real data and the intermediate representation as the result of the encoder network:

$$\begin{aligned}\phi : X &\mapsto F \\ \psi : F &\mapsto X\end{aligned}$$

We can define the intermediate representation computation and the reconstruction of the input data with the following equations:

$$z = \sigma(Wx + b), \quad x \sim X \in \mathbb{R}^d, \quad z \sim F \in \mathbb{R}^p \quad (2.28)$$

$$x' = \sigma'(W'z + b') \quad (2.29)$$

The primal training objective for the autoencoders is the reconstruction loss. Depending on the architecture of the decoder or the desired task, the source of the parameters for the objective function can change. Autoencoder learns the representation of the data by minimizing the reconstruction loss.

$$\phi, \psi = \operatorname{argmin}_{\phi, \psi} \|X - (\psi \circ \phi)X\| \quad (2.30)$$

$$L(x, x') = \|x - x'\|^2 = \|x - \sigma'(W'\sigma(Wx + b) + b')\|^2 \quad (2.31)$$

With enough model capacity and training time, autoencoder might learn to copy the input perfectly. This reduces the functionality of the encoder-decoder modules to a identity function:

$$x \sim X \in \mathbb{R}^d, \quad \psi(\phi(x)) = x$$

This is generally an unwanted outcome. To overcome this problem, autoencoders are designed to learn the representation of the data imperfectly. [18] This is achieved by how the encoder module learns to compress the data to a meaningful representation. These 3 types of autoencoder models can give an idea about how the architecture can be changed to alter the amount of information that can be extracted from the data. These are:

- Undercomplete Autoencoders
- Regularized Autoencoders
- Denoising Autoencoders

2. STATE OF THE ART

The undercomplete autoencoders reduce the dimension size of the latent representation. Learning purposefully insufficient representation forces autoencoder to keep the most salient features about the data. The training objective function is the reconstruction loss which presented in equation 2.30. If the computation of this function is linear, then the autoencoder mimics the behaviour of a principal component analysis method. When the encoder transition ϕ and the decoder transition ψ is designed to have a nonlinear nature, then the autoencoder has the ability to capture more complex representation of the data, more powerful approach than the PCA. But this has the risk of again, causing autoencoder to replicate the input data perfectly and becoming an identity function $x' = \psi(\phi(x))$. Regularized autoencoders' design difference is that their latent representation layer usually has same or higher dimensional power (more neurons from the perspective of multilayer perceptrons). To acquire a similar performance like undercomplete variants, they regularize the loss function with the potential properties of the data distribution such as robustness to the noise or the sparsity of the representation. Prioritizing which aspects of the data will be used as a regularizer term for the objective function is a task dependent problem. [18]. Denoising autoencoders tries to solve a different problem to achieve a good representation.

$$\hat{x} = x + \epsilon : x \sim X \in \mathbb{R}^d, \quad \epsilon \sim \mathcal{N}(\mu, \sigma^2)$$

The input data is corrupted with some kind of noise, gaussian etc. Denoising autoencoders tries to reconstruct the data and undo this corruption by minimizing this updated loss with the difference of the \hat{x}

$$L(x, \psi(\phi(\hat{x})))$$

2.3 GAN Based Anomaly Detection Methods

Anomaly detection on spatial data using reconstruction based methods primarily converges around variational autoencoders and other stuff. The GAN based approaches in this field are relatively new. In the following subsections, I will explain the approaches that can be considered as the state of the art for this problem in the order of their complexity and incremental improvements. Each approach will be evaluated by their problem description, the type of data they use, model architecture and anomaly score computation.

2.3.1 AnoGAN

AnoGAN model is the first approach that utilizes GANs for the anomaly detection task. [42] It focuses on the problem of detecting anomalous regions in optical coherence tomography images. Images containing retinal fluid or hyperreflective foci (indicator of disease progression in various retinal diseases) are identified as anomalous examples. The main motivation to use GANs to obtain a model which represents normal anatomical variability of the data is that previous supervised approaches involves training a model with a large dataset using annotated examples of known markers. Relying on vocabulary of markers may limit the predictive power of the model as the image contains much more relevant information than the marker and extensive supervised training is a problem because it means that every stage of the disease requires an extensive training with annotated examples such as labeled lesions which may decrease limit of exploiting imaging data for treatment decisions. [42].

The framework consists of 2 parts. First part is the GAN module, generator and discriminator networks. Their architecture is inspired by the original DCGAN architecture[36]. The objective function of the GAN is the same of the equation 2.3. During the training the generator enhances its ability to generate more realistic retinal images while the discriminator learns the diagnose the real and the generated sample more efficiently.

The second part of the framework is about the inverse mapping of the noise to the image. We know from the GANs training procedure that when the adversarial training is completed, the generator network has learnt the distribution p_g that is theoretically very similar to the input data distribution p_{data} . (I state "theoretically" explicitly because depending on the configuration of the GAN and the data that is used, this may differ significantly in practise. This aspect of the learning will be mentioned in the latter approaches.) Even though with the trained generator,

$$G(z) = z \mapsto x, x \sim p_{\text{data}}$$

the GAN framework does not learn the inverse mapping from the image to the noise. AnoGAN framework aims to find the noise sample z^* from the query image (normal or anomalous) such that the generated image $G(z^*)$ is visually the most similar one to the query image. This is essential for the anomaly detection task. z^* is sampled randomly as initialization. Then the $G(z^*)$ is computed and based on the defined loss function gradients the coefficients of z^* is updated using backpropagation. This procedure is repeated for a number of epochs to obtain the final value of the z^* .

2. STATE OF THE ART

The loss function for the inverse mapping is defined as the combination of two separate losses. Residual loss is responsible for enforcing visual similarity between the generated image $G(z\gamma)$ and the query image x .

$$L_R(z_\gamma) = \sum \|x - G(z_\gamma)\| \quad (2.32)$$

Discrimination loss on the other hand enforces the generated image $G(z\gamma)$ to lie on the manifold X of image x .

$$L_D(z_\gamma) = \sum \|f(x) - f(G(z_\gamma))\| \quad (2.33)$$

There are two different implementation for the discrimination loss. One with sigmoid cross entropy which is from the original GAN implementation [19] and one with feature matching [40] which uses the last layer of the discriminator rather than the output to compute the loss. Using this loss allows them to use discriminator not as a classifier but as a feature extractor. Using both residual and discrimination loss, the total loss function is defined as :

$$L(z_\gamma) = (1 - \lambda) \times L_R(\mathbf{z}_\theta) + \lambda \times L_D(\mathbf{z}_\gamma)$$

Only the coefficients of z is adapted via backpropagation. The trained weights of the generator and discriminator are fixed in this stage. [42]

The Anomaly detection task uses same loss functions to compute the anomaly score.

$$A(x) = (1 - \lambda) \times R(x) + \lambda \times D(x)$$

The $R(x)$ is the residual score value and the $D(x)$ is the discrimination score value at the last update iteration of the inverse mapping procedure, respectively. The model outputs a larger anomaly score $A(x)$ for anomalous samples though smaller score means that very similar image is seen during the training of the GAN.

This model is the first model that integrates generative adversarial networks to an anomaly detection framework. The mapping from latent representation to image data is completely dependent on the architecture of the GAN and its training parameters. The disadvantageous aspect of this approach is the inference. For each query image framework approximates the latent representation vector using backpropagation steps which makes the whole process significantly slow compared to the other approaches. Improvements to this framework is discussed in the Latest Developments chapter (3.1). The experiment results and the implementation details will be covered in chapter 5.

2.3.2 BiGAN and ALI

BiGAN (Bidirectional Generative adversarial Networks) [13] and ALI (Adversarially Learned Inference) [14] are two models that independent from each other but focuses on the same problem. They investigate to acquire efficient inference method from data to latent space by using encoders in the adversarial training framework of GANs. They emphasize learning the inverse mapping for the auxiliary supervised tasks such as classification, segmentation and discrimination. The only difference between the models is that the BiGAN uses a deterministic network for the encoder while ALI uses a stochastic network. I will first briefly explain the original architecture of the ALI and continue to present the framework in detail using BiGAN instead since their explanation for the model is more consistent with the GAN framework. Furthermore I will discuss how this architecture can be used for anomaly detection. In the original papers, anomaly detection has not been explored as an auxiliary supervised task but the later models like in section 2.3.3 or in section 2.3.4, used these architectures as a state of the art comparison in terms of both foundation and performance measure hence their use of anomaly detection will also be discussed.

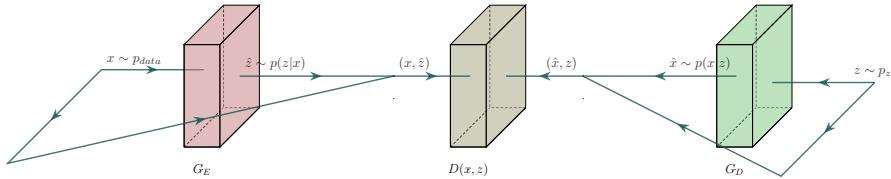


Figure 24 – ALI architecture overview

Aligan architecture is depicted as a generator and a discriminator in figure 24. The significant change in ALI is that the generator is the combination of decoder network and an encoder network. Encoder represents the joint distribution $p_E(x, z)$ which has the marginal $p_{\text{data}}(x)$ and decoder represents the joint distribution $p_D(x, z)$ with the marginal $p_z(z)$ which represents the factorized noise distribution (generally $\mathcal{N}(0, 1)$). The objective function of the ALI is to match these two joint distributions. Successing in this goal also ensures the match of the marginals and the conditional distributions $p(z|x)$ and $p(x|z)$. Discriminator is also modified to support this new objective function. It is trained to distinguish between the joint pairs $(x, \hat{z} = G_E(x))$ and $(\hat{x} = G_D(z), z)$ as data and noise.

BiGAN architecture model is presented in figure 25. The architectures are theoretically identical with the exception of the encoder network choice

2. STATE OF THE ART

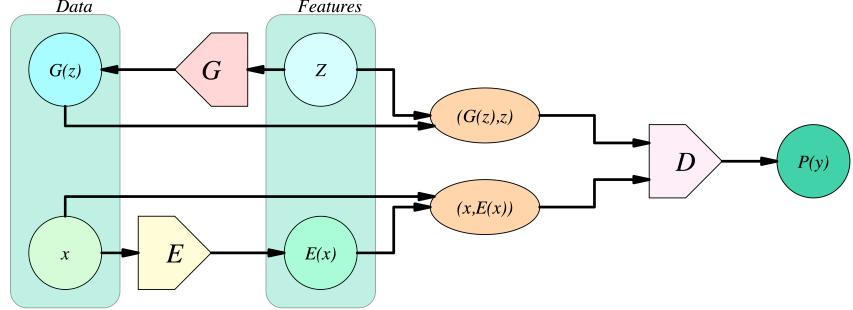


Figure 25 – BiGAN architecture overview

in practise. BiGAN experiments are conducted with a deterministic network whilst ALI framework used a stochastic network. Apart from the generator, BiGAN also trains the encoder network in an adversarial manner. Same objective function used in GAN provides the relationship between the generator and encoder that $G = E^{-1}$ and vice versa. This invertibility is the key aspect of the model and the other frameworks to provide an efficient inference with reliable sampling. We Know from the section 2.2.1 that:

$$G : \Omega_z \mapsto \Omega_x$$

We define the Encoder with the reverse of that transition:

$$E : \Omega_x \mapsto \Omega_z$$

Since this framework accepts the networks as deterministic we also redefine the distributions for generator and encoder networks:

$$\begin{aligned} p_G(x|z) &= \delta(x - G(z)) \\ p_E(z|x) &= \delta(z - E(x)) \end{aligned}$$

The objective function of the framework can be characterized as:

$$\min_{G,E} \max_D V(D, E, G) \quad (2.34)$$

where

$$V(D, E, G) := \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{X}}} \left[\underbrace{\mathbb{E}_{\mathbf{z} \sim p_E(\cdot|\mathbf{x})} [\log D(\mathbf{x}, \mathbf{z})]}_{\log D(\mathbf{x}, E(\mathbf{x}))} \right] +$$

2.3. GAN Based Anomaly Detection Methods

$$\mathbb{E}_{\mathbf{z} \sim p_{\mathbf{Z}}} \underbrace{\left[\mathbb{E}_{\mathbf{x} \sim p_G(\cdot | \mathbf{z})} [\log(1 - D(\mathbf{x}, \mathbf{z}))] \right]}_{\log(1 - D(G(\mathbf{z}), \mathbf{z}))} \quad (2.35)$$

To prove that the encoder is the inverse of the generator, optimal discriminator needs to be determined.[13] The joint distributions of the generator and encoder can be illustrated as the marginals of the data and the noise and related conditional probability distributions:

$$p_{GZ}(x, z) := p_G(x|z)p_Z(z) \quad (2.36)$$

$$p_{EX}(x, z) := p_E(z|x)p_X(x) \quad (2.37)$$

The joint latent space and the data for the problem can also be illustrated as a joint representation.

$$\Omega := \Omega_X \times \Omega_Z$$

For the region $R \subseteq \Omega$ probability measures[25] of encoder and generator is defined in equation 2.38 and 2.41 respectively.[13]

$$P_{EX}(R) := \int_{\Omega} p_{EX}(\mathbf{x}, \mathbf{z}) \mathbf{1}^2_{[(\mathbf{x}, \mathbf{z}) \in R]} d(\mathbf{x}, \mathbf{z}) = \quad (2.38)$$

$$= \int_{\Omega_X} p_{\mathbf{X}}(\mathbf{x}) \int_{\Omega_Z} p_E(\mathbf{z}|\mathbf{x}) \mathbf{1}_{[(\mathbf{x}, \mathbf{z}) \in R]} d\mathbf{z} d\mathbf{x} \quad (2.39)$$

$$= \int_{\Omega_X} p_{\mathbf{X}}(\mathbf{x}) \mathbf{1}_{[(\mathbf{x}, E(\mathbf{x})) \in R]} d\mathbf{x} \quad (2.40)$$

$$P_{GZ}(R) := \int_{\Omega} p_{GZ}(\mathbf{x}, \mathbf{z}) \mathbf{1}_{[(\mathbf{x}, \mathbf{z}) \in R]} d(\mathbf{x}, \mathbf{z}) = \quad (2.41)$$

$$= \int_{\Omega_Z} p_{\mathbf{Z}}(\mathbf{z}) \int_{\Omega_X} p_G(\mathbf{x}|\mathbf{z}) \mathbf{1}_{[(\mathbf{x}, \mathbf{z}) \in R]} d\mathbf{x} d\mathbf{z} \quad (2.42)$$

$$= \int_{\Omega_Z} p_{\mathbf{Z}}(\mathbf{z}) \mathbf{1}_{[(G(\mathbf{z}), \mathbf{z}) \in R]} d\mathbf{z} \quad (2.43)$$

² Indicator function is a type of characteristic function that is used to indicate a membership on a subset X . It is denoted with bold $\mathbf{1}$ symbol.

2. STATE OF THE ART

Probability measures over regions on the image data $R_X \subseteq \Omega_X$ and the latent noise $R_Z \subseteq \Omega_Z$ can be similarly defined.

$$P_{\mathbf{X}}(R_{\mathbf{X}}) := \int_{\Omega_{\mathbf{X}}} p_{\mathbf{X}}(\mathbf{x}) \mathbf{1}_{[\mathbf{x} \in R_{\mathbf{X}}]} d\mathbf{x} \quad (2.44)$$

$$P_{\mathbf{X}}(R_{\mathbf{X}}) := \int_{\Omega_{\mathbf{X}}} p_{\mathbf{X}}(\mathbf{x}) \mathbf{1}_{[\mathbf{x} \in R_{\mathbf{X}}]} d\mathbf{x} \quad (2.45)$$

The optimal discriminator follows the same derivation from the GAN framework [19]. The Objective equation 2.35 can be reduced to the Jensen-Shannon divergence between joint distributions $P_{E\mathbf{X}}$ and $P_{G\mathbf{Z}}$ with the proof of the following propositions from the BiGAN [13].

Proposition 2.3.1. *For any E and G , the optimal discriminator $D_{EG}^* := \underset{D}{\text{argmax}} V(D, E, G)$ is the Radon-Nikodym derivative $f_{EG} := \frac{dP_{E\mathbf{X}}}{dP_{E\mathbf{X}} + dP_{G\mathbf{Z}}} : \Omega \mapsto [0, 1]$ of measure $P_{E\mathbf{X}}$ with respect to measure $dP_{E\mathbf{X}} + dP_{G\mathbf{Z}}$.*

Proposition 2.3.2. *The encoder and generator's objective for an optimal discriminator $C(E, G) := \underset{D}{\text{max}} V(D, E, G) = V(D_{EG}^*, E, G)$ can be rewritten in terms of the Jensen-Shannon divergence between the measures $P_{E\mathbf{X}}$ and $P_{G\mathbf{Z}}$ as $C(E, G) = 2D_{JS}(P_{E\mathbf{X}} \| P_{G\mathbf{Z}}) - \log 4$*

Discriminator is trained with both the encoder and generator network output. Therefore it is rational to assume that if we were to define a probability measure for discriminator . So we let P_{EG} be the average of the $P_{E\mathbf{X}}$ and $P_{G\mathbf{Z}}$.

$$P_{EG} = \frac{P_{E\mathbf{X}} + P_{G\mathbf{Z}}}{2} \quad (2.46)$$

Since both $P_{E\mathbf{X}}$ and $P_{G\mathbf{Z}}$ are continuous with respect to the P_{EG} , the relationship of RN derivaties of f_{EG} and f_{GE} is defined accordingly:

$$f_{EG} := \frac{dP_{E\mathbf{X}}}{dP_{E\mathbf{X}} + dP_{G\mathbf{Z}}} \quad (2.47)$$

$$f_{GE} := \frac{dP_{G\mathbf{Z}}}{dP_{E\mathbf{X}} + dP_{G\mathbf{Z}}} \quad (2.48)$$

$$f_{EG} + f_{GE} := \frac{dP_{G\mathbf{Z}}}{dP_{E\mathbf{X}} + dP_{G\mathbf{Z}}} + \frac{dP_{E\mathbf{X}}}{dP_{E\mathbf{X}} + dP_{G\mathbf{Z}}} = \frac{d(P_{G\mathbf{Z}} + P_{E\mathbf{X}})}{d(P_{G\mathbf{Z}} + P_{E\mathbf{X}})} = 1 \quad (2.49)$$

2.3. GAN Based Anomaly Detection Methods

Using the property of RN derivative (equation 2.24) we can transform the expected value function and rewrite the objective function of the BiGAN with a single probability measure.[13] [19]

$$\mathbb{E}_{\mathbf{x} \sim P}[g(\mathbf{x})] = \int_{\Omega} g dP = \int_{\Omega} g \frac{dP}{dQ} dQ = \int_{\Omega} g f_{PQ} dQ = \mathbb{E}_{\mathbf{x} \sim Q} [f_{PQ}(\mathbf{x}) g(\mathbf{x})] \quad (2.50)$$

We start by rewriting the original objective function with the joint probability distributions.

$$V(D, E, G) = \mathbb{E}_{(\mathbf{x}, \mathbf{z}) \sim P_{E\mathbf{X}}} [\log D(\mathbf{x}, \mathbf{z})] + \mathbb{E}_{(\mathbf{x}, \mathbf{z}) \sim P_{G\mathbf{Z}}} [\log(1 - D(\mathbf{x}, \mathbf{z}))] \quad (2.51)$$

Then the transformation described in equation 2.50 is applied.

$$= \mathbb{E}_{(\mathbf{x}, \mathbf{z}) \sim P_{EG}} [\underbrace{2f_{EG}(\mathbf{x}, \mathbf{z}) \log D(\mathbf{x}, \mathbf{z})}_{\frac{dP_{EX}}{dP_{EG}}} + \underbrace{\mathbb{E}_{(\mathbf{x}, \mathbf{z}) \sim P_{EG}} [2f_{GE}(\mathbf{x}, \mathbf{z}) \log(1 - D(\mathbf{x}, \mathbf{z}))]}_{\frac{dP_{GZ}}{dP_{EG}}}] \quad (2.52)$$

The final stage of the equation falls into a type of function $f(a, y) = a \log y + (1-a) \log(1-y)$ [13] which provides $\underset{y}{\operatorname{argmax}} f(a, y) = a$ for any $a \in [0, 1]$. Thus the optimal Discriminator $D_{EG}^* = f_{EG}$. That proves the first proposition.

$$= 2\mathbb{E}_{(\mathbf{x}, \mathbf{z}) \sim P_{EG}} [f_{EG}(\mathbf{x}, \mathbf{z}) \log D(\mathbf{x}, \mathbf{z}) + f_{GE}(\mathbf{x}, \mathbf{z}) \log(1 - D(\mathbf{x}, \mathbf{z}))] \quad (2.53)$$

$$= 2\mathbb{E}_{(\mathbf{x}, \mathbf{z}) \sim P_{EG}} [f_{EG}(\mathbf{x}, \mathbf{z}) \log D(\mathbf{x}, \mathbf{z}) + (1 - f_{EG}(\mathbf{x}, \mathbf{z})) \log(1 - D(\mathbf{x}, \mathbf{z}))] \quad (2.54)$$

Second proposition can be proved with the same approach from GAN[19] using the connection acquired from the first proposition. [13]

First the objective equation is rewritten with the optimal discriminator and its equivalent RN derivative. [13]

$$C(E, G) = \max_D V(D, E, G) = V(D_{EG}^*, E, G) \quad (2.55)$$

$$= \mathbb{E}_{(\mathbf{x}, \mathbf{z}) \sim P_{E\mathbf{X}}} [\log D_{EG}^*(\mathbf{x}, \mathbf{z})] + \mathbb{E}_{(\mathbf{x}, \mathbf{z}) \sim P_{G\mathbf{Z}}} [\log(1 - D_{EG}^*(\mathbf{x}, \mathbf{z}))] \quad (2.56)$$

$$= \mathbb{E}_{(\mathbf{x}, \mathbf{z}) \sim P_{E\mathbf{X}}} [\log f_{EG}(\mathbf{x}, \mathbf{z})] + \mathbb{E}_{(\mathbf{x}, \mathbf{z}) \sim P_{G\mathbf{Z}}} [\log f_{GE}(\mathbf{x}, \mathbf{z})] \quad (2.57)$$

Then addition - subtraction of $\log 4$ is applied following the equation 2.21 to interpret the result as Jensen-Shannon divergence, hence proving

2. STATE OF THE ART

the second proposition.

$$= \mathbb{E}_{(\mathbf{x}, \mathbf{z}) \sim P_{E\mathbf{X}}} [\log(2f_{EG}(\mathbf{x}, \mathbf{z}))] + \mathbb{E}_{(\mathbf{x}, \mathbf{z}) \sim P_{G\mathbf{Z}}} [\log(2f_{GE}(\mathbf{x}, \mathbf{z}))] - \log 4 \quad (2.58)$$

$$= D_{KL}(P_{E\mathbf{X}} \| P_{EG}) + D_{KL}(P_{G\mathbf{Z}} \| P_{EG}) - \log 4 \quad (2.59)$$

$$= D_{KL}\left(P_{E\mathbf{X}} \parallel \frac{P_{E\mathbf{X}} + P_{G\mathbf{Z}}}{2}\right) + D_{KL}\left(P_{G\mathbf{Z}} \parallel \frac{P_{E\mathbf{X}} + P_{G\mathbf{Z}}}{2}\right) - \log 4 \quad (2.60)$$

$$= 2D_{JS}(P_{E\mathbf{X}} \| P_{G\mathbf{Z}}) - \log 4. \square \quad (2.61)$$

The proposition 2.3.2 allows us to characterize the objective function of the BiGAN in terms of the Jensen-Shannon divergence but implicitly it also proves just like in equation 2.26 that the global minimum of the training criterion is only achieved with the equality of the probability measures $P_{E\mathbf{X}}$ and $P_{G\mathbf{Z}}$. Consequentially this gives the optimality condition for both encoder and generator.

Theorem 2.3.1. *if E and G are an optimal encoder and generator, then $E = G^{-1}$ almost everywhere; that is, $G(E(x)) = x$ for P_X almost every $x \in \Omega_x$, and $E(G(z)) = z$ for P_z , almost every $z \in \Omega_z$. [13]*

The inversion property of the BiGAN framework is dependent on the factor of $P_{E\mathbf{X}}$ being equal to $P_{G\mathbf{Z}}$ in the optimal state. Let R_X^0 be the region that the reconstructed image from the encoded noise is not equal to the input image and let R^0 be the region for the data, noise pairs which the data is also a member of the R_X^0 . Using the optimal encoder and decoder, it can be proven that the probability measure for the region R_X^0 is zero, meaning that for almost every x in the region Ω_x , $G(E(x)) = x$ and vice versa.

$$R_X^0 := \{x \in \Omega_x : x \neq G(E(x))\} \quad (2.62)$$

$$R^0 := \{(x, z) \in \Omega : \mathbf{z} = E(x) \wedge x \in R_X^0\} \quad (2.63)$$

$$P_{\mathbf{X}}(R_X^0) = \int_{\Omega_{\mathbf{X}}} p_{\mathbf{X}}(\mathbf{x}) \mathbf{1}_{[\mathbf{x} \in R_X^0]} d\mathbf{x} \quad (2.64)$$

$$= \int_{\Omega_{\mathbf{X}}} p_{\mathbf{X}}(\mathbf{x}) \mathbf{1}_{[(\mathbf{x}, E(\mathbf{x})) \in R^0]} d\mathbf{x} \quad (2.65)$$

$$= P_{E\mathbf{X}}(R^0) \quad (2.66)$$

2.3. GAN Based Anomaly Detection Methods

$$= P_{GZ}(R^0) \quad (2.67)$$

$$= \int_{\Omega_Z} p_Z(z) \mathbf{1}_{[(G(z), z) \in R^0]} dz \quad (2.68)$$

$$= \int_{\Omega_Z} p_Z(z) \mathbf{1}_{[z = E(G(z)) \wedge G(z) \in R_X^0]} dz \quad (2.69)$$

$$= \int_{\Omega_Z} \underbrace{p_Z(z) \mathbf{1}_{[z = E(G(z)) \wedge G(z) \neq G(E(G(z)))]}}_{=0 \text{ for any } z, \text{ as } z=E(G(z)) \Rightarrow G(z)=G(E(G(z)))} dz \quad (2.70)$$

$$= 0. \square \quad (2.71)$$

$$(2.72)$$

Training of BiGAN framework is implemented using the same adversarial process of GANs. Encoder and generator network are trained together since they both try to minimize the objective. Each iteration is divided into two parts. First the discriminator is trained using the outputs from the encoder and generator with fixed weights, pushing the gradient of the function towards a positive step. Also the parameters of discriminator θ_D is updated. Then θ_E and θ_G the parameters of encoder and generator are updated concurrently with a negative direction in the gradient of the objective function. [13]

Loss functions for the training optimization and Anomaly score computation differs in terms of the type of approach, different from AnoGAN[42]. AnoGAN framework used the combination of the reconstruction loss (equation 2.32) and the discrimination loss (equation 2.33) both to infer z from the query image and to compute the anomaly score. BiGAN framework's encoder is trained with the kind of function that is used to train the generator, sigmoid cross entropy. The advantage of the BiGAN is the reduced inference time to compute the anomaly score because the inferred noise from the query image \hat{x} is the output from the generator $\hat{z} = E(\hat{x})$.

Anomaly score computation of BiGAN however, is the same approach used in AnoGAN. It uses a combination of the discriminaton loss and a reconstruction loss. The degree of norms used in the computation differs in the experiment stage to observe the behavior of the anomaly score. The Anomalous samples of the test dataset is determined by accepting the anomaly score that is greater then a predetermined threshold. This varies depending on the type of dataset and the dimensional complexity of the dataset the framework trained on.

2.3.3 ALAD

Adversarially Learned Anomaly Detection, or ALAD [48] is a framework designed for the anomaly detection task. It builds upon the idea of AnoGAN, using the base architecture from the AliGAN and BiGAN with additional improvements incorporated to increase the stabilization of the adversarial training in both discriminator module and encoder, generator pair. I will first describe the improvements introduced to the adversarial training framework and discuss their potential advantages over the previous approach. Then I will explain the ALAD architecture in detail. Lastly, learning of the model and the anomaly score computation choices will be discussed.

2.3.3.1 Spectral Normalization

Even though the training of GAN works in practice, it may easily become unstable in certain conditions. For example, in high dimensional space, density ratio estimation of the discriminator network is often inaccurate and this results in generator networks fail to learn the dimensional structure of the target data distribution. [5] Consequently, when the support of the model distribution and the target distribution is disjoint, then the discriminator may become decently efficient classifying real data from the generated data sample. In this scenario, the generator fails to learn to improve sample fake image data. Spectral Normalization is one of the recent proposed improvement methods to stabilize the weights of the discriminator network by bounding its gradients using Lipschitz continuity. [32]

Consider the optimal discriminator of $D(\mathbf{x}, \theta) = \mathcal{A}(f(\mathbf{x}, \theta))$ in equation 2.16. It takes the form :

$$D_G^*(\mathbf{x}) = \frac{q_{\text{data}}(\mathbf{x})}{q_{\text{data}}(\mathbf{x}) + p_G(\mathbf{x})} = \text{sigmoid}(f^*(\mathbf{x})),$$

where $f^*(\mathbf{x}) = \log q_{\text{data}}(\mathbf{x}) - \log p_G(\mathbf{x}) \quad (2.73)$

Where \mathcal{A} is an activation function corresponding to the divergence of distance measure of the user's choice (usually a sigmoid). With the derivative of:

$$\nabla_{\mathbf{x}} f^*(\mathbf{x}) = \frac{1}{q_{\text{data}}(\mathbf{x})} \nabla_{\mathbf{x}} q_{\text{data}}(\mathbf{x}) - \frac{1}{p_G(\mathbf{x})} \nabla_{\mathbf{x}} p_G(\mathbf{x}) \quad (2.74)$$

2.3. GAN Based Anomaly Detection Methods

This gradient function can be unbounded or even incomputable depending on the distribution. Spectral normalization is applied to introduce regularity condition to derivative $f(\mathbf{x})$. [32]

Discriminator function can be modified to apply this regularization to the objective function:

$$\arg \max_{\|f\|_{\text{Lip}} \leq K} V(G, D) \quad (2.75)$$

Where $\|f\|_{\text{Lip}}$ is depicted as the smallest value M such that:

$$\|f(\mathbf{x}) - f(\mathbf{x}')\| / \|\mathbf{x} - \mathbf{x}'\| \leq M \quad (2.76)$$

for any x, x' with the norm being the l_2 norm. x_2 in X ,[43]

Using the computation of the K-Lipschitz function of the each layer in discriminator, the spectral norm of each layer weight matrix can be normalized so the updated weights of the discriminator after each epoch of training becomes:

$$\bar{W}_{\text{SN}}(W) := W / \sigma(W) \quad (2.77)$$

2.3.3.2 Conditional Entropy (ALICE Framework)

The BiGAN framework in section 2.3.2 aims to learn both the mapping from the input image data to a latent representation with its encoder module and the inverse of that mapping relation with its generator module. It acquires these mappings by training its enoder and generator modules together in an adversarial manner to learn a joint distribtuion depicted in equation 2.37 and 2.36. The problem suggested by the ALICE framework[29] is that learning the joint distribution alone doesn't help the model to reconstruct images necesarily faithful to the reproductions of the input data. [29] This is because the objective function is built only to match joint distributions of image and noise pairs at the same time potential dependency structures or correlations between two random variables within each joint repsentation is not specified or constrained. [29]

In order to reduce the identifiability issues associated with the objective of BiGAN, the conditionals $P_G(x|z)$ and $P_E(z|x)$ needs to be constrained and be under supervision using a paired samples from the domain Ω_x and

2. STATE OF THE ART

Ω_z . Information-theoretic measure conditional entropy is used to provide this regularization.

Conditional entropy quantifies the amount of information needed to describe the outcome of a random variable Y given that the value of another random variable X is known [12] under the joint distribution

$$\pi(x, y), \quad x \in X, y \in Y$$

For the BiGAN framework the conditional entropies are defined below:

$$H^\pi(\mathbf{x}|\mathbf{z}) \triangleq -\mathbb{E}_{\pi(\mathbf{x}, \mathbf{z})}[\log \pi(\mathbf{x}|\mathbf{z})] \quad (2.78)$$

$$H^\pi(\mathbf{z}|\mathbf{x}) \triangleq -\mathbb{E}_{\pi(\mathbf{x}, \mathbf{z})}[\log \pi(\mathbf{z}|\mathbf{x})] \quad (2.79)$$

The Conditional entropy defined above is dependant on the underlying distribution of the random variables, in this case the distribution of the encoder and generator modules $p_G(x)$ and $p_E(z)$. In practise this value is intractable because we dont have access to the saddle points of the objective function beforehand. As an alternative, the approximation of the conditional entropy is calculated by bounding CE to the criterion of cycle-consistency [50]

Cycle consistency is the similarity of the reconstruction to the input data. Denoting the reconstruction of x as x' , the cycle can be defined as:

$$x' = G(z'), \quad z' = E(x), \quad x \in \Omega_x, \quad x' \approx G(E(x)) \quad (2.80)$$

In other gan frameworks like Cycle-GAN[50], Disco-GAN [24] and Dual-GAN [47] cycle consistency constraint is implemented leveraging ℓ_k losses with $k = 1, 2$. Using ℓ_2 based pixel-wise loss functions to reconstruct the samples leads to blurry images during inference [27][29]. To enforce a better constraint and to improve the quality of the reconstructed samples, this framework suggests using the feature layer of the discriminator to compute the loss for the conditional entropy constraint. The objective function to approximate this constraint is given below:

$$\begin{aligned} \min_{E, G} \max_D \mathcal{L}_{\text{Cycle}}(E, G, D) &= \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log \sigma(f_D(\mathbf{x}, \mathbf{x}))] \\ &\quad + \mathbb{E}_{\hat{\mathbf{x}} \sim p_G(\hat{\mathbf{x}}|\mathbf{z}), \mathbf{z} \sim p_E(\mathbf{z}|\mathbf{x})} \log (1 - \sigma(f_D(\mathbf{x}, \hat{\mathbf{x}}))) \end{aligned} \quad (2.81)$$

2.3. GAN Based Anomaly Detection Methods

This additional adversarial training is implemented into the ALAD architecture to enforce a constraint to improve the quality of reconstructed noise and the image. Its addition is explained in the next part of the section.

2.3.3.3 ALAD Architecture

Training of the framework shares the same adversarial approach with BiGAN (2.3.2). Generator module simultaneously learns the mapping from input data to a latent representation while the encoder module learns the opposite. Model also incorporates the mentioned improvements to increase the stabilization of the training procedure. The architecture overview is illustrated in figure 26

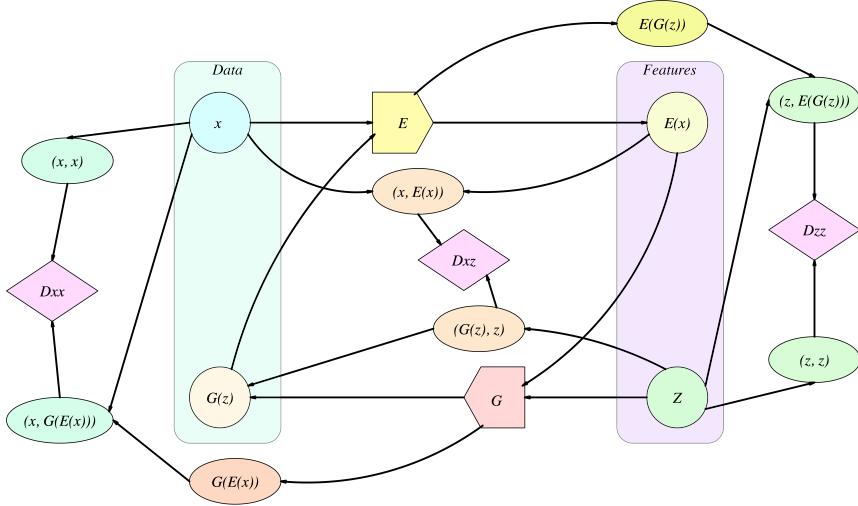


Figure 26 – ALAD architecture overview

Formally the ALAD architecture tries to match the joint distributions

$$p_E(x, z) = p_E(z|x)p_{\text{data}}(x)p_G(x, z) = p_G(x|z)p_Z(z) \quad (2.82)$$

and tries to separate the data, noise pairs $(x, E(x))$ and $(G(z), z)$ with an adversarial discriminator module D. For this framework we refer to the main discriminator as D_{xz} . The core objective function with the updated terms is :

$$\begin{aligned} V(D_{xz}, E, G) &= \mathbb{E}_{x \sim p_{\text{data}}} [\log D_{xz}(x, E(x))] \\ &\quad + \mathbb{E}_{z \sim p_Z} [1 - \log D_{xz}(G(z), z)] \end{aligned} \quad (2.83)$$

2. STATE OF THE ART

Section 2.3.3.2 explains the convergence problem of the joint distribution. Because of the lack of constraints imposed on the p_{data} and p_Z , the objective function may not converge properly and this consequentially decreases the quality of the reconstructed samples, namely $x' \approx G(E(x))$. In addition to the image reconstruction, to enforce a constraint on the noise encoding from the input image, $z' \approx (E(G(z)))$, ALAD architecture also adds a second objective function to the overall adversarial training loop to regularize the conditional distributions. The CE term for the image and for the noise is defined below.

$$H^\pi(x|z) = -\mathbb{G}_{\pi(x,z)}[\log \pi(x|z)] \quad (2.84)$$

$$H^\pi(z|x) = -\mathbb{E}_{\pi(x,z)}[\log \pi(z|x)] \quad (2.85)$$

As explained in the previous section, since the saddle points beforehand can't be obtained for the computation, these terms can be approximated with the discriminators D_{xx} and D_{zz} respectively. The training of these discriminators is added to the main adversarial training loop as a part of the discriminator module training. Their objective functions are defined in equation 2.86

$$\begin{aligned} V(D_{xx}, E, G) &= \mathbb{E}_{x \sim p_{\text{data}}} [\log D_{xx}(x, x)] \\ &\quad + \mathbb{E}_{x \sim p_{\text{data}}} [1 - \log D_{xx}(x, G(E(x)))] \end{aligned} \quad (2.86)$$

$$\begin{aligned} V(D_{zz}, E, G) &= \mathbb{E}_{z \sim p_Z} [\log D_{zz}(z, z)] \\ &\quad + \mathbb{E}_{z \sim p_Z} [1 - \log D_{zz}(z, E(G(z)))] \end{aligned} \quad (2.87)$$

Putting it all together, ALAD framework solves the following combination of value function during the adversarial training.

$$\begin{aligned} \min_{G,E} \max_{D_{xz}, D_{xx}, D_{zz}} V(D_{xz}, D_{xx}, D_{zz}, E, G) &= \\ V(D_{xz}, E, G) + V(D_{xx}, E, G) + V(D_{zz}, E, G) \end{aligned} \quad (2.88)$$

During the training, the generator and the encoder are updated simultaneously. In the second part of the adversarial training, all the discriminators update their weights to both regularize the reconstructions and to converge to equilibrium for the main value function. Spectral normalization is used in every convolutional layer of the discriminators. Framework also implemented them to the layers of the encoder network with the purpose of further increasing the quality of the reconstructed samples. [48]. The result of the training and its analysis will be discussed in chapter 5.

2.3. GAN Based Anomaly Detection Methods

The Anomaly score computation of the framework is based on the quality of the reconstructed query samples. If the reconstructed query image is distant to the original input it will obtain a greater anomaly score. ALAD framework suggests that because of the nature of the spatial data, even though there are similarities, obtained noise during reconstruction may mask potential feature similarities, preventing their detection. The suggested approach of the framework is to compute the anomaly score using the feature layer of the D_{xx} discriminator shown in equation 2.89.

$$A(x) = \|f_{xx}(x, x) - f_{xx}(x, G(E(x)))\|_1 \quad (2.89)$$

In this equation, $f(\cdot, \cdot)$ represents the vector of activations of the layer before the output of the D_{xx} discriminator [48]. Anomaly score $A(\cdot)$ captures the confidence such that the given image is well encoded and reconstructed by the generator and therefore it is from the real data distribution. [48] However different kinds of anomaly score computations are also considered for the sake of comparison.

These are (including the one we discussed):

$$\begin{aligned} \bullet L_1 : A(x) &= \|x - x'\|_1 \\ \bullet L_2 : A(x) &= \|x - x'\|_2 \\ \bullet \text{Logits} : A(x) &= \log(D_{xx}(x, x')) \\ \bullet \text{Features} : A(x) &= \|f_{xx}(x, x) - f_{xx}(x, x')\|_1 \end{aligned} \quad (2.90)$$

Overall this framework aims to improve the generator and encoder learning with improvements such as spectral normalization and conditional entropy discriminators. Its performance with our dataset and ways to improve the adversarial training will be discussed in chapter 5

2.3.4 GANomaly & Skip-GANomaly

GANomaly framework [2] and Skip-GANomaly [3] are the most recent models that focuses on the anomaly detection using adversarial training. Their performance on the benchmark datasets like CIFAR-10 [26] and SVHN [34] exceeds the previous mentioned frameworks. This section will explain their structure and their differences in terms of one another.

GANomaly and Skip-GANomaly models differ with a significant change in the model itself. Both models contain a generator and discriminator

2. STATE OF THE ART

modules but generator module is actually composed of an adversarial autoencoder network. Details of the both architecture will be discussed in order.

Generator module of the GANomaly framework composed of an encoder-decoder network as can be seen in figure 27. The third module is the encoder network for the generated samples. Generator network takes the image data as an input and sequentially it creates the latent representation of the image and reconstruction from the latent representation then the second encoder network encodes the latent representation of the reconstructed sample. Discriminator network shares the same functionality as in GAN framework. [19]

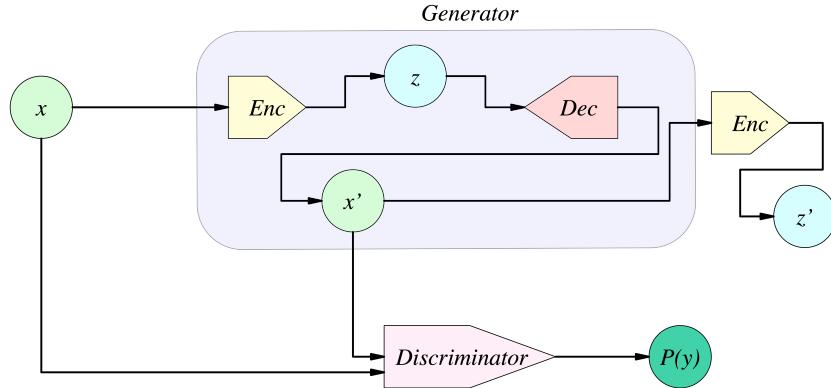


Figure 27 – GANomaly architecture overview

With the use of the adversarial autoencoders, framework obtains a superior reconstruction ability compared to the previous models. But in doing so, its generative sub-module decoder is not a true generative component from the perspective of the generative adversarial networks. With the change of the model architecture its training objective also changes. GANomaly framework uses adversarial training scheme but its generator module loss is the combination of 3 separate loss functions. These are:

- Adversarial Loss
- Contextual Loss
- Encoder Loss

Adversarial loss represents the loss function component for the adversarial training of the framework. Instead of maximizing the probability

2.3. GAN Based Anomaly Detection Methods

of discriminator's attaining generated image as real, it has a loss function derived from the [40] that uses feature matching. The main purpose of this type of loss function is to reduce the instability in the GAN training. Adversarial loss depicted in equation 2.91 is the \mathcal{L}_2 distance of the feature representation of the original and the generated image, respectively. [2] Feature layer is extracted from the activation layer before the output of the discriminator.

$$\mathcal{L}_{adv} = \mathbb{E}_{x \sim p_{\mathbf{x}}} \|f(x) - \mathbb{E}_{x \sim p_{\mathbf{x}}} f(G(x))\|_2 \quad (2.91)$$

Only adversarial loss does not help to optimize the generator towards reconstructing images similar to the input data. To this end, contextual loss is also added to the generator loss function. According to [21], using \mathcal{L}_1 instead of \mathcal{L}_2 decreases the blurriness in the generated images, so the following contextual loss that measures the distance between original image and the generated one is added to the total generator loss.

$$\mathcal{L}_{con} = \mathbb{E}_{x \sim p_{\mathbf{x}}} \|x - G(x)\|_1 \quad (2.92)$$

Additional encoder loss is also added to the generator to improve the encoding of the latent representation of the reconstruction. This loss function aims to minimize the distance between the encoded latent representation of the input image and the reconstruction. The Loss function is depicted below.

$$\mathcal{L}_{enc} = \mathbb{E}_{x \sim p_{\mathbf{x}}} \|G_E(x) - E(G(x))\|_2 \quad (2.93)$$

Overall, the objective loss function of the generator is defined in equation 2.94 where the weight parameters w_{adv}, w_{con} and w_{enc} are used to change the impact of individual losses.

$$\mathcal{L} = w_{adv}\mathcal{L}_{adv} + w_{con}\mathcal{L}_{con} + w_{enc}\mathcal{L}_{enc} \quad (2.94)$$

GANomaly framework uses encoder loss function also as an anomaly score measure.

$$\mathcal{A}(\hat{x}) = \|G_E(\hat{x}) - E(G(\hat{x}))\|_1 \quad (2.95)$$

Skip-GANomaly framework[3] is the continuation of the GANomaly architecture with changes mainly to the generator network. As can be seen from figure 28, encoder and decoder networks inside the generator framework is connected using skip connections.

2. STATE OF THE ART

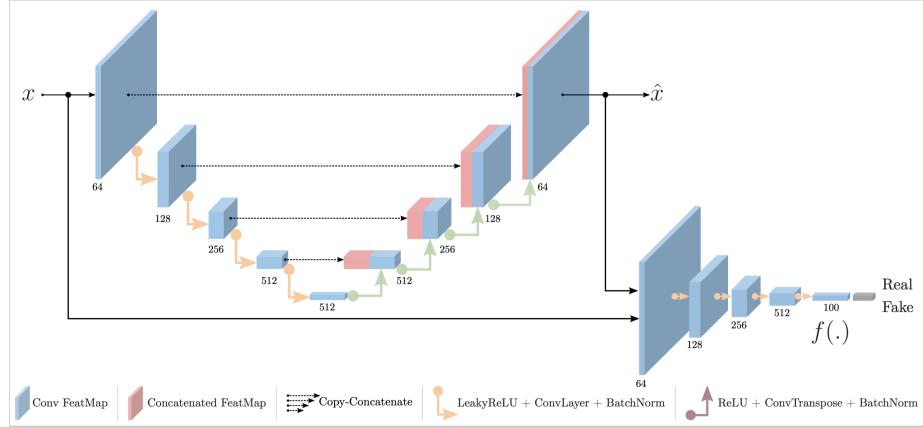


Figure 28 – Skip GANomaly architecture overview [3]

At each layer of the encoder, convolutional feature map is concatenated to the corresponding decoder layer. Addition of skip connections provides a better reconstruction capability to the generator than the GANomaly framework. [2]. Generator objective function is built using the same logic. Because of the exclusion of the secondary encoder network, the encoder loss is obtained using the feature layer of the discriminator.

$$\mathcal{L}_{enc} = \mathbb{E}_{x \sim p_x} |f(x) - f(\hat{x})|_2 \quad (2.96)$$

To find anomalies during the inference stage, Skip-GANomaly framework combines the encoder loss with the use of contextual loss.

$$\mathcal{A}(\dot{x}) = \lambda R(\dot{x}) + (1 - \lambda) L(\dot{x}) \quad (2.97)$$

$R(\dot{x})$ represents the contextual loss with the reconstruction term and $L(\dot{x})$ represents the encoder loss with the latent representation term. Term λ is used to adjust the impact of the loss functions to the overall anomaly score.

Both GANomaly and Skip-GANomaly uses adversarial encoder architecture in their generator networks and therefore have superior reconstruction capabilities compared to the previous 3 framework. The primal problem with their architecture is that they don't adapt the true distribution of the input data. They learn the underlying latent representation well to reconstruct the input and detect anomalies. However their generator network represents the scenario where the generator has learnt the input data distribution fairly well. In order to improve the results of a GAN based anomaly

2.3. GAN Based Anomaly Detection Methods

detection framework, insights gained from these 2 networks are significantly important. Their contribution to the improved framework and their performance analysis regards to other models will be explored in chapters 4 and 5 respectively.

CHAPTER 3

Latest Developments

This chapter presents the developments in the field of anomaly detection and generative adversarial networks. Although the models mentioned in the state of the art chapter deliver certain level of performance, potential improvements are still a possibility. These improvements are related to both adversarial training of the generator and discriminator networks and different strategy to stabilize the training of encoder network present in the overall framework. In the following sections, some of these improvements that are later adopted by our framework will be introduced and their contribution to solve the disadvantages we observed in the previous frameworks will be discussed.

3.1 F-AnoGAN Framework

AnoGAN [42] is considered as the first framework that uses generative adversarial networks for an anomaly detection task. The main problems with this framework, are the stabilization issues of the adversarial training, and the second stage of the framework which was mapping from the latent representation to the input data distribution which can be also considered as the inverse mapping of the generator. The model used back propagation to approximate the latent representation for every query image to compute the anomaly score which resulted a very poor performance in terms of the computation time. This was the main disadvantage of the framework because it is very challenging to integrate into a real life application with an implausible inference time. F-AnoGAN (Fast AnoGAN) framework [41] aims to eliminate this issue by implementing a new training strategy. It

3. LATEST DEVELOPMENTS

also uses a new objective function for the adversarial training to further stabilize the generator discriminator performance. In the rest of the section, F-AnoGAN framework, its training strategies and anomaly detection methods will be discussed.

The framework architecture of F-AnoGAN is very similar to the BiGAN [13] framework. It consists of a generator discriminator for the adversarial learning and an encoder network to learn the inverse mapping from latent representation to the input image data.

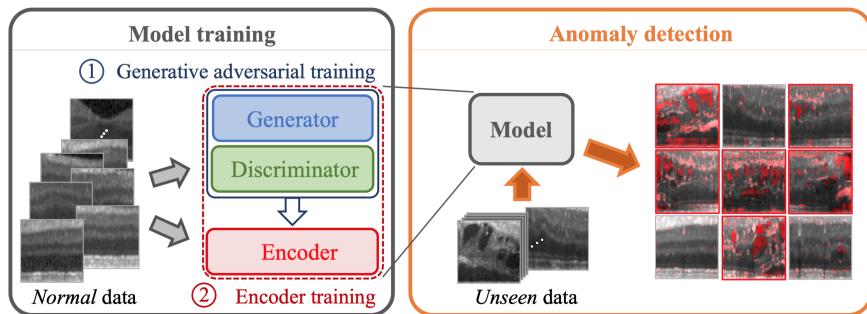


Figure 31 – F-AnoGAN Framework Overview [41]

The first improvement is the change of the training style of the whole framework. In BiGAN approach, encoder and generator is trained simultaneously to fool the discriminator. Hence discriminator is modified to classify the pairs of noise (latent representation) and images instead of the single image approach used in the GAN framework. Therefore it tries to distinguish samples from a joint distribution which explained in section 2.3.2. Including encoder to the adversarial setup also introduces instability issues which ALAD Framework [48] tried to mitigate with additional discriminators that approximate conditional entropy. (section 2.3.3.2). F-AnoGAN framework addresses this issue by separating the training of the encoder from the generator discriminator pair. The new framework can be seen in figure 32.

This new training scheme comprises of the stages. In the first stage GAN framework is trained. The objective function for the adversarial training is also replaced with the Wasserstein GAN's objective function [6] although training of the GAN can be performed with any other framework's approach (including the original GAN framework) according to the [41].

In the second stage, encoder network is trained using generator and discriminator with locked weight. Two separate pipelines are used to measure

3.1. F-AnoGAN Framework

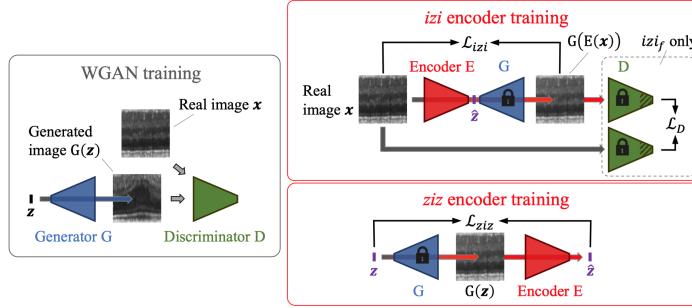


Figure 32 – F-AnoGAN Training Strategies [41]

the performance of the encoder. These are *izi* (image-noise-image) and *ziz* (noise-image-noise) respectively.

IZI Encoder Training

IZI method follows the standard autoencoder network approach. The generator network with fixed weights acts as a decoder network. During the training, same image dataset used in the first stage for training gan is mapped to a latent mapping z by a trainable encoder and then reconstructed using the fixed generator network. Training objective for this setup is to minimize MSE (Mean Squared Error) based reconstruction error of the input image x and the reconstructed image $G(E(x))$.

$$\mathcal{L}_{izi}(\mathbf{x}) = \frac{1}{n} \|\mathbf{x} - G(E(\mathbf{x}))\|^2 \quad (3.1)$$

This approach has an important drawback regarding to latent representation. Since the distribution of the latent representation of the input image is not known, the encoder is trained only using a form of contextual loss which enforce the similarity only in the image space. Without the insufficient information about the latent space, encoder may map images to a representations such that the reconstructions are not convincing enough for the discriminator to evaluate as real. [41]. Therefore the framework suggests also including a latent space based loss function derived from the feature layer of the discriminator to guide the encoder training. Improved izi_f training objective is defined below.

$$\mathcal{L}_{izi_f}(\mathbf{x}) = \frac{1}{n} \cdot \|\mathbf{x} - G(E(\mathbf{x}))\|^2 + \frac{\kappa}{n_d} \cdot \|f(\mathbf{x}) - f(G(E(\mathbf{x})))\|^2 \quad (3.2)$$

where the $f(\cdot)$ represents the feature layer of the discriminator as a statistics, n_d is the dimensionality of the intermediate feature representation

3. LATEST DEVELOPMENTS

f and κ denotes the weighting factor for the inclusion of the discriminator guidance.

ZIZ Encoder Training

Different from the *IZI* method, this approach reverses the auto encoder setup and forms a decoder encoder architecture. A random sampled noise from the z-space is mapped to the image space using the fixed generator and then generated sample is encoded using the trainable encoder network. The loss for the training is defined as the MSE based reconstruction of the noise which depicted in equation below.

$$\mathcal{L}_{ziz}(\mathbf{z}) = \frac{1}{d} \|\mathbf{z} - E(G(\mathbf{z}))\|^2 \quad (3.3)$$

The shortcoming of this approach is that even though the encoder is trained with a loss that will enforce a similarity in latent space, encoder sees only images that are generated by the generator. It doesn't see any image samples from the training dataset which affects the contextual similarity of the reconstruction of learned latent space distribution.

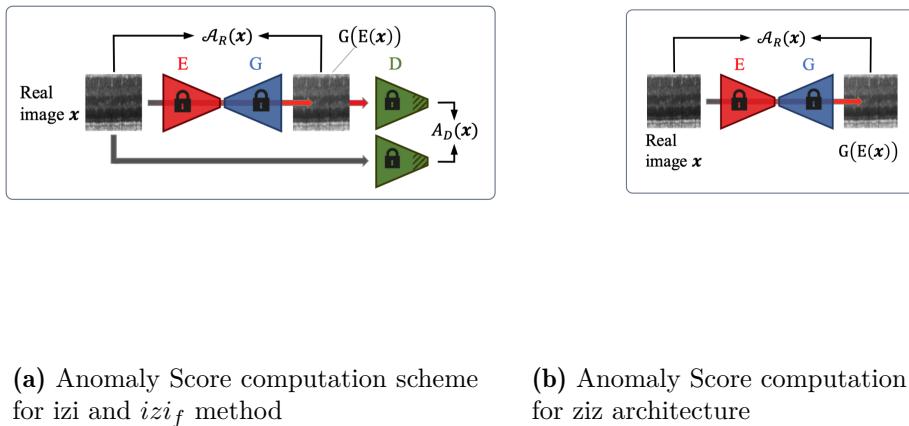


Figure 33 – Anomaly Score computations for both training methods [41]

To calculate the anomaly score for inference, deviation of the query images from their reconstructions are quantified. Figure 33 represents the anomaly score computations for both training methods.

3.2. Energy Based Generative Adversarial Networks

IZI_f method uses the same loss functions for the computation of the anomaly score. Anomaly score for image x is defined as:

$$\mathcal{A}(\mathbf{x}) = \mathcal{A}_R(\mathbf{x}) + \kappa \cdot \mathcal{A}_D(\mathbf{x}) \quad (3.4)$$

$$\mathcal{A}_R(\mathbf{x}) = \frac{1}{n} \cdot \|\mathbf{x} - G(E(\mathbf{x}))\|^2 \quad (3.5)$$

$$\mathcal{A}_D(\mathbf{x}) = \frac{1}{n_d} \cdot \|f(\mathbf{x}) - f(G(E(\mathbf{x})))\|^2 \quad (3.6)$$

For IZI and ZIZ method, the computation of the anomaly score reduces down to $\mathcal{A}_R(\mathbf{x})$ function only. Both training methods yields a similar anomaly score computation for the anomalous samples. Since the models are trained with a dataset that doesn't contain any anomalies, the query images that have anomalous regions results in poorer reconstruction hence a higher anomaly score while the normal images produce a more similar reconstructions to the original sample.

The Significance of this framework is that it separates the training of the encoder from the adversarial setting of the GAN's while preserving the inverse mapping functionality for the inference. Chapter 4 will explain its contribution to the proposed improved framework.

3.2 Energy Based Generative Adversarial Networks

Training a GAN framework perhaps the most sensitive part of working with them. There has been an extensive amount of research to improve the stabilization of the adversarial training or perhaps offer a new methodology. [40] and [5] offer number modifications to the original adversarial training setup to improve the generation performance ,decrease the stabilization and prevent issues such as mode collapse¹. [6] and [20] offers a new method to compute the loss and ways to stabilize it. Mainly because of its adversarial setting, training of GAN frameworks will always be a topic of interest. Energy based generative adversarial networks [49] offers a new training method by redefining the loss functions and the functionality of the discriminator network. This section will introduce the architecture and how it works.

¹Mode collapse is the scenerio where generator learns enough information to fool the discriminator very early. It finds a generation that fools the discriminator and

3. LATEST DEVELOPMENTS

Energy based models measures the current state of the model by assigning a scalar predefined energy as a degree of compatibility [28]. Training energy based models consists of mapping low energy outputs to the samples with the "correct" class, and mapping higher energy outputs to the "incorrect" class. From the perspective of GANs, EBGAN model is trained to output lower energy for the real data, while the generated data (fake images) outputs a higher energy. The model architecture can be seen in figure 34.

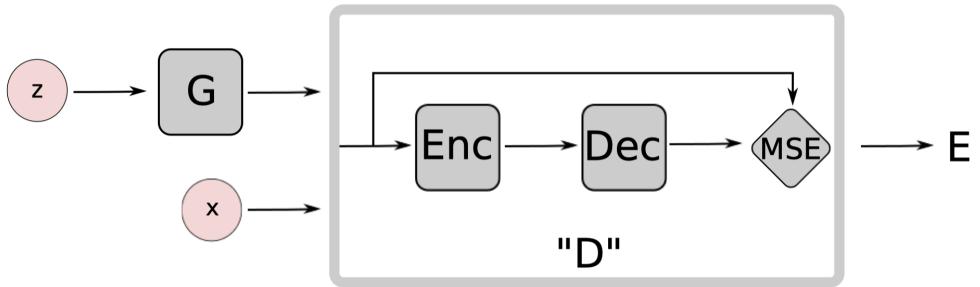


Figure 34 – EBGAN Model Structure [49]

The main structural difference of this framework from the other ones is that discriminator is defined as an auto encoder network to provide the reconstruction of the inputs. The energy value function is defined as the margin loss of the input and the reconstructed sample [49] as a reconstruction error.

Formally a margin is used to create an energy gap between the correct output and the incorrect output. For data sample x , and a generated sample $G(z)$ with z being sampled from a known distribution p_z , The discriminator loss \mathcal{L}_D and the generator loss \mathcal{L}_G are formally defined in equation 3.7.

$$\begin{aligned}
 D(x) &= \| \text{Dec}(\text{Enc}(x)) - x \| \\
 \mathcal{L}_D(x, z) &= D(x) + [m - D(G(z))]^+ \\
 \mathcal{L}_G(z) &= D(G(z))
 \end{aligned} \tag{3.7}$$

where $D(x)$ is the reconstruction loss and $[\cdot]^+ = \max(0, \cdot)$. Minimizing \mathcal{L}_G with respect to the generator is similar to maximizing the second term of the \mathcal{L}_D . Instead of calculating the probability of the input being real or

does not learn anything else from the discriminator. Because of that all the generated samples from the generator look the same, defeating the purpose of generating from the distribution of the data.[5]

3.2. Energy Based Generative Adversarial Networks

fake, discriminator is responsible for attaining the correct amount of energy to the input images.

Using autoencoder network as a discriminator brings additional advantages to the training setting. Rather than using a single target information to train the model, reconstruction based output offers a more diverse targets for the discriminator [49]. One of the reasons of the stabilization issues in the adversarial training is that discriminator might provide gradients not useful enough for generator to improve the quality of the generated samples. Having a reconstruction loss output might provide a better gradient in terms of different directions as opposed to binary logistic loss. Another advantage is that even though they are trained with an adversarial setting, autoencoder networks might prove useful to learn the underlying energy manifold of the data without supervision or negative examples when they are trained using a proper regularization constraint. This means that in this setting generated samples and input data are both used to model the underlying energy manifold of the data.

EBGAN framework argues that, the discriminator network is regularized by exposing generated samples from the generator which it should output higher reconstruction energies [49]. The benefit of this regularization is that instead of a hand crafted predefined one, the regularization term can also be trained along with the discriminator. Adversarial training allows a direct connection between learning the energy function and producing contradictive samples in that regard.

EBGAN framework also adds additional penalty called pulling away term (PT) that run at a representation level to prevent mode collapse depicted in equation 3.8.

$$f_{PT}(S) = \frac{1}{N(N-1)} \sum_i \sum_{j \neq i} \left(\frac{S_i^\top S_j}{\|S_i\| \|S_j\|} \right)^2 \quad (3.8)$$

S is the feature output from the encoder for generated images. Pulling-away term measures the cosine similarity among all generated images features S in a minibatch. If the mode collapses, the feature vectors will thereupon be similar, i.e. the angles are close to zero and the cosine will max-out. Therefore, it will add a high penalty if there are too similar. [49]

CHAPTER 4

Architectural Improvements

This chapter presents

- 4.1 SEM Images Dataset**
- 4.2 Analysis of Aforementioned Approaches**
- 4.3 Encoded Energy Based Generative Adversarial Network**
- 4.4 Sequentially Encoded Energy Based Generative Adversarial Network**

CHAPTER 5

Experimental Results

5.1 Experiment Settings

5.1.1 Figure of Merit

5.2 GAN Based Model Analysis

test

TABLE 51
ABLATION STUDY FOR ANOGAN TO TEST THE EFFECT OF VARIOUS
TRAINING IMPROVEMENTS FOR STABILIZATION.

Model	Metrics				
	AUROC	Precision	Recall	F1 Score	
AnoGAN	Normal				
	IN				
	SL				
	LF				
	IN + SL				
	IN + LF				
	SL + LF				
	LF + SL + IN				

5. EXPERIMENTAL RESULTS

TABLE 52
ABLATION STUDY FOR BiGAN TO TEST THE EFFECT OF VARIOUS TRAINING IMPROVEMENTS FOR STABILIZATION.

Model	Metrics			
	AUROC	Precision	Recall	F1 Score
BiGAN	Normal			
	IN			
	SL			
	LF			
	IN + SL			
	IN + LF			
	SL + LF			
	LF + SL + IN			

TABLE 53
ABLATION STUDY FOR ALAD TO TEST THE EFFECT OF VARIOUS TRAINING IMPROVEMENTS FOR STABILIZATION.

Model	Metrics			
	AUROC	Precision	Recall	F1 Score
ALAD	Normal			
	IN			
	SL			
	LF			
	IN + SL			
	IN + LF			
	SL + LF			
	LF + SL + IN			

5.3. AR Based Model Analysis

TABLE 54

ABLATION STUDY FOR GANOMALY TO TEST THE EFFECT OF VARIOUS TRAINING IMPROVEMENTS FOR STABILIZATION.

Model	Metrics			
	AUROC	Precision	Recall	F1 Score
GANomaly	Normal			
	IN			
	SL			
	LF			
	IN + SL			
	IN + LF			
	SL + LF			
	LF + SL + IN			

TABLE 55

ABLATION STUDY FOR SKIP-GANOMALY TO TEST THE EFFECT OF VARIOUS TRAINING IMPROVEMENTS FOR STABILIZATION.

Model	Metrics			
	AUROC	Precision	Recall	F1 Score
Skip-GANomaly	Normal			
	IN			
	SL			
	LF			
	IN + SL			
	IN + LF			
	SL + LF			
	LF + SL + IN			

5.3 AR Based Model Analysis

5.4 Improved Model Analysis

5.5 Discussion of Experiment Results

This chapter will explain the experimental results for all the models and improvements. It will also point out a discussion about what could be improved and the future direction.

There will be 3 classes of models to be considered.

- Models that maps z to x to find the image distribution and uses

5. EXPERIMENTAL RESULTS

- inverse sample for reconstruction → AnoGAN, BiGAN and ALAD
- Models that maps directly image distribution by integrating an encoder to the generator module hence creating in practise an autoencoder, and uses again, reconstruction → Ganomaly and skip ganomaly
- Models that tries new methods to explore different solution strategies
 - Training with full image → Segmentation papers
 - Ganomaly + Noise addition (one Class paper concurrent work) to improve the performance of the image distribution learning.
 - Energy Based GANS (loss function change), Can this method be applied to the encoder network as well, to better capture the noise distribution.
- All models will be tested with the standard improvements of the gan training.
 - Label Flipping → improves gradints of the discriminator
 - Soft Labels → Better than 0,1 reference the papers
 - Adding noise to the input image and rectracted image to confuse discriminator → robustness
- Best performance models then will be tested with a training with validation that is based on the reconstruction of the image.
- All model results with ablation study
- Improved model results
- Visual Results like precision recall, AUC curve, Histogram of Anomalies
- Numerical Results like the result of the model performances in a table ‘†’

CHAPTER 6

Conclusion and Future Work

This chapter will explain the purpose of the thesis once more and give information about the general process and the result interpretation (1 page)

It will make suggestions based on the improved model experiments and leave an open door for the improvement.

APPENDIX A

Model Implementation Details

A.1 AnoGAN Implementation

anogan placeholder

TABLE A1
ARCHITECTURE AND HYPERPARAMETERS OF ANOGAN MODEL

Operation	Kernel	Stride	Feature Maps/ Units	BN ?	Activation
Generator					
Dense			$4 \times 4 \times 512$	✓	Leaky Relu
Transposed Convolution	4×4	2×2	512	✓	Leaky Relu
Transposed Convolution	4×4	2×2	256	✓	Leaky Relu
Transposed Convolution	4×4	2×2	128	✓	Leaky Relu
Transposed Convolution	5×5	1×1	1		TanH
Latent Dimension	256				
Leaky Relu Slope	0.2				
Batch Norm Momentum	0.8				
Optimizer	Adam (lr = $5e - 5$, beta 1 = 0.5, beta 2 = 0.999,)				
Discriminator					
Convolution	4×4	2×2	64	✓	Leaky Relu
Convolution	4×4	2×2	128	✓	Leaky Relu
Convolution	4×4	2×2	256	✓	Leaky Relu
Dense			1		Sigmoid
Leaky ReLU Slope	0.2				
Batch Norm Momentum	0.8				
Optimizer	Adam (lr = $1e - 6$, beta 1 = 0.5, beta 2 = 0.999,)				
Epochs	50				
Batch Size	64				

A. MODEL IMPLEMENTATION DETAILS

TABLE A2
ARCHITECTURE AND HYPERPARAMETERS OF BiGAN MODEL

Operation	Kernel	Stride	Feature Maps/ Units	BN ?	Activation
Generator					
Dense			$4 \times 4 \times 512$	✓	Leaky Relu
Transposed Convolution	4×4	2×2	512	✓	Leaky Relu
Transposed Convolution	4×4	2×2	256	✓	Leaky Relu
Transposed Convolution	4×4	2×2	128	✓	Leaky Relu
Transposed Convolution	$\times 5$	1×1	1		TanH
Latent Dimension	256				
Leaky Relu Slope	0.2				
Batch Norm Momentum	0.8				
Optimizer	Adam (lr = 5e - 5, beta 1 = 0.5, beta 2 = 0.999,)				
Discriminator					
Convolution	4×4	2×2	64	✓	Leaky Relu
Convolution	4×4	2×2	128	✓	Leaky Relu
Convolution	4×4	2×2	256	✓	Leaky Relu
Dense			1		Sigmoid
Leaky ReLU Slope	0.2				
Batch Norm Momentum	0.8				
Optimizer	Adam (lr = 1e - 6, beta 1 = 0.5, beta 2 = 0.999,)				
Encoder					
Convolution	4×4	2×2	64	✓	Leaky Relu
Convolution	4×4	2×2	128	✓	Leaky Relu
Convolution	4×4	2×2	256	✓	Leaky Relu
Dense			256		
Leaky ReLU Slope	0.2				
Batch Norm Momentum	0.8				
Optimizer	Adam (lr = 1e - 6, beta 1 = 0.5, beta 2 = 0.999,)				
Epochs	50				
Batch Size	64				

A.2 BiGAN Implementation

bigan placeholder

A.3 ALAD Implementation

ALAD placeholder

A.4 GANomaly Implementation

Ganomaly placeholder

A.4. GANomaly Implementation

TABLE A3
ARCHITECTURE AND HYPERPARAMETERS OF ALAD MODEL

Operation	Kernel	Stride	Feature Maps/ Units	BN ?	Activation
Generator					
Dense			4 × 4 × 512	✓	Leaky Relu
Transposed Convolution	4×4	2×2	512	✓	Leaky Relu
Transposed Convolution	4×4	2×2	256	✓	Leaky Relu
Transposed Convolution	4×4	2×2	128	✓	Leaky Relu
Transposed Convolution	4×4	1×1	1		TanH
Latent Dimension	256				
Leaky Relu Slope	0.2				
Batch Norm Momentum	0.8				
Optimizer	Adam (lr = 5e - 5, beta 1 = 0.5, beta 2 = 0.999,)				
Discriminator XZ					
Convolution(x)	4×4	2×2	128	✓	Leaky Relu
Convolution(x)	4×4	2×2	256	✓	Leaky Relu
Convolution(x)	4×4	2×2	512	✓	Leaky Relu
Reshape	Batch × 512 × 4 × 4				
Convolution(z)	4×4	2×2	512	Dropout	Leaky Relu
Convolution(z)	4×4	2×2	512	Dropout	Leaky Relu
Concatenate					
Convolution	1×1	1×1	1024	Dropout	Leaky Relu
Convolution	1×1	1×1	1024	Dropout	Leaky Relu
Dense			1		Sigmoid
Leaky ReLU Slope	0.2				
Dropout Rate	0.2				
Batch Norm Momentum	0.8				
Optimizer	Adam (lr = 1e - 6, beta 1 = 0.5, beta 2 = 0.999,)				
Encoder					
Convolution	4×4	2×2	32	✓	Leaky Relu
Convolution	4×4	2×2	64	✓	Leaky Relu
Convolution	4×4	2×2	128	✓	Leaky Relu
Convolution	4×4	2×2	256	✓	Leaky Relu
Convolution	4×4	2×2	256		
Leaky ReLU Slope	0.2				
Batch Norm Momentum	0.8				
Optimizer	Adam (lr = 1e - 6, beta 1 = 0.5, beta 2 = 0.999,)				
Discriminator XX					
Concatenate	f				
Convolution	4×4	2×2	64	Dropout	Leaky Relu
Convolution	4×4	2×2	128	Dropout	Leaky Relu
Dense			1		Sigmoid
Leaky ReLU Slope	0.2				
Batch Norm Momentum	0.8				
Optimizer	Adam (lr = 1e - 6, beta 1 = 0.5, beta 2 = 0.999,)				
Discriminator ZZ					
Dense	4×4	2×2	64	Dropout	Leaky Relu
Dense	4×4	2×2	32	Dropout	Leaky Relu
Dense	4×4	2×2	1	Dropout	Leaky Relu
Leaky ReLU Slope	0.2				
Dropout Rate	0.2				
Batch Norm Momentum	0.8				
Optimizer	Adam (lr = 1e - 6, beta 1 = 0.5, beta 2 = 0.999,)				
Epochs	50				59
Batch Size	64				

A. MODEL IMPLEMENTATION DETAILS

TABLE A4
ARCHITECTURE AND HYPERPARAMETERS OF GANOMALY MODEL

Operation	Kernel	Stride	Feature Maps/ Units	BN ?	Activation
Generator					
Encoder					
Convolution	5×5	2×2	64		Leaky Relu
Convolution	5×5	2×2	128	✓	Leaky Relu
Convolution	5×5	2×2	256	✓	Leaky Relu
Dense			256		
Decoder					
Transposed Convolution	4×4	2×2	256	✓	Leaky Relu
Transposed Convolution	4×4	2×2	128	✓	Leaky Relu
Transposed Convolution	4×4	2×2	64	✓	Leaky Relu
Transposed Convolution	4×4	2×2	32	✓	Leaky Relu
Transposed Convolution	4×4	2×2	16	✓	Leaky Relu
Transposed Convolution	4×4	1×1	1		TanH
Latent Dimension	256				
Leaky Relu Slope	0.2				
Batch Norm Momentum	0.8				
Optimizer	Adam (lr = 5e - 5, beta 1 = 0.5, beta 2 = 0.999,)				
Encoder 2					
Convolution	5×5	2×2	64		Leaky Relu
Convolution	5×5	2×2	128	✓	Leaky Relu
Convolution	5×5	2×2	256	✓	Leaky Relu
Dense			256		
Discriminator					
Convolution	4×4	2×2	64	✓	Leaky Relu
Convolution	4×4	2×2	128	✓	Leaky Relu
Convolution	4×4	2×2	256	✓	Leaky Relu
Dense			1		Sigmoid
Leaky ReLU Slope	0.2				
Batch Norm Momentum	0.8				
Optimizer	Adam (lr = 1e - 6, beta 1 = 0.5, beta 2 = 0.999,)				
Epochs	50				
Batch Size	64				

A.5. Skip-GANomaly Implementation

TABLE A5
ARCHITECTURE AND HYPERPARAMETERS OF SKIP-GANOMALY MODEL

Operation	Kernel	Stride	Feature Maps/ Units	BN ?	Activation
Generator					
Dense			4 × 4 × 512	✓	Leaky Relu
Transposed Convolution	4×4	2×2	512	✓	Leaky Relu
Transposed Convolution	4×4	2×2	256	✓	Leaky Relu
Transposed Convolution	4×4	2×2	128	✓	Leaky Relu
Transposed Convolution	4×4	1×1	1		TanH
Latent Dimension	256				
Leaky Relu Slope	0.2				
Batch Norm Momentum	0.8				
Optimizer	Adam (lr = 5e - 5, beta 1 = 0.5, beta 2 = 0.999,)				
Discriminator					
Convolution	4×4	2×2	64	✓	Leaky Relu
Convolution	4×4	2×2	128	✓	Leaky Relu
Convolution	4×4	2×2	256	✓	Leaky Relu
Dense			1		Sigmoid
Leaky ReLU Slope	0.2				
Batch Norm Momentum	0.8				
Optimizer	Adam (lr = 1e - 6, beta 1 = 0.5, beta 2 = 0.999,)				
Epochs	50				
Batch Size	64				

A.5 Skip-GANomaly Implementation

A.6 EBGAN Implementation

A.7 SENCEBGAN Implementation

A. MODEL IMPLEMENTATION DETAILS

TABLE A6
ARCHITECTURE AND HYPERPARAMETERS OF EBGAN MODEL

Operation	Kernel	Stride	Feature Maps/ Units	BN ?	Activation
Generator					
Dense			$4 \times 4 \times 512$	✓	Leaky Relu
Transposed Convolution	4×4	2×2	512	✓	Leaky Relu
Transposed Convolution	4×4	2×2	256	✓	Leaky Relu
Transposed Convolution	4×4	2×2	128	✓	Leaky Relu
Transposed Convolution	4×4	1×1	1		TanH
Latent Dimension	256				
Leaky Relu Slope	0.2				
Batch Norm Momentum	0.8				
Optimizer	Adam (lr = 5e - 5, beta 1 = 0.5, beta 2 = 0.999,)				
Discriminator					
Convolution	4×4	2×2	64	✓	Leaky Relu
Convolution	4×4	2×2	128	✓	Leaky Relu
Convolution	4×4	2×2	256	✓	Leaky Relu
Dense			1		Sigmoid
Leaky ReLU Slope	0.2				
Batch Norm Momentum	0.8				
Optimizer	Adam (lr = 1e - 6, beta 1 = 0.5, beta 2 = 0.999,)				
Epochs	50				
Batch Size	64				

TABLE A7
ARCHITECTURE AND HYPERPARAMETERS OF SENCEEBGAN MODEL

Operation	Kernel	Stride	Feature Maps/ Units	BN ?	Activation
Generator					
Dense			$4 \times 4 \times 512$	✓	Leaky Relu
Transposed Convolution	4×4	2×2	512	✓	Leaky Relu
Transposed Convolution	4×4	2×2	256	✓	Leaky Relu
Transposed Convolution	4×4	2×2	128	✓	Leaky Relu
Transposed Convolution	4×4	1×1	1		TanH
Latent Dimension	256				
Leaky Relu Slope	0.2				
Batch Norm Momentum	0.8				
Optimizer	Adam (lr = 5e - 5, beta 1 = 0.5, beta 2 = 0.999,)				
Discriminator					
Convolution	4×4	2×2	64	✓	Leaky Relu
Convolution	4×4	2×2	128	✓	Leaky Relu
Convolution	4×4	2×2	256	✓	Leaky Relu
Dense			1		Sigmoid
Leaky ReLU Slope	0.2				
Batch Norm Momentum	0.8				
Optimizer	Adam (lr = 1e - 6, beta 1 = 0.5, beta 2 = 0.999,)				
Epochs	50				
Batch Size	64				

Bibliography

- [1] Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. “A survey of network anomaly detection techniques”. In: *Journal of Network and Computer Applications* 60 (2016), pp. 19 –31. ISSN: 1084-8045. DOI: <https://doi.org/10.1016/j.jnca.2015.11.016>. URL: <http://www.sciencedirect.com/science/article/pii/S1084804515002891>.
- [2] Samet Akçay, Amir Atapour Abarghouei, and Toby P. Breckon. “GANomaly: Semi-supervised Anomaly Detection via Adversarial Training”. In: *ACCV*. 2018.
- [3] Samet Akçay, Amir Atapour Abarghouei, and Toby P. Breckon. “Skip-GANomaly: Skip Connected and Adversarially Trained Encoder-Decoder Anomaly Detection”. In: *CoRR* abs/1901.08954 (2019).
- [4] Plamen Angelov, Ramin Ramezani, and Xiaowei Zhou. “Autonomous Novelty Detection and Object Tracking in Video Streams using Evolving Clustering and Takagi-Sugeno type Neuro-Fuzzy System”. In: June 2008, pp. 1456–1463. DOI: 10.1109/IJCNN.2008.4633989.
- [5] Martin Arjovsky and Léon Bottou. “Towards Principled Methods for Training Generative Adversarial Networks”. In: *stat* 1050 (Jan. 2017).
- [6] Martín Arjovsky, Soumith Chintala, and Léon Bottou. “Wasserstein GAN”. In: *CoRR* abs/1701.07875 (2017).
- [7] M. Behniafar, A.R. Nowroozi, and H.R. Shahriari. “A Survey of Anomaly Detection Approaches in Internet of Things”. In: *The ISC International Journal of Information Security* 10.2 (2018), pp. 79–92. ISSN: 2008-2045. DOI: 10.22042/isecure.2018.116976.408. eprint: http://www.isecure-journal.com/article__835f5ca77e495cec328b957653e1f4d566995.pdf. URL: http://www.isecure-journal.com/article_66995.html.
- [8] Yoshua Bengio. “Learning Deep Architectures for AI”. In: *Found. Trends Mach. Learn.* 2.1 (Jan. 2009), pp. 1–127. ISSN: 1935-8237. DOI: 10.1561/2200000006. URL: <http://dx.doi.org/10.1561/2200000006>.

BIBLIOGRAPHY

- [9] Patrick Billingsley. *Probability and Measure*. Second. John Wiley and Sons, 1986.
- [10] Varun Chandola, Arindam Banerjee, and Vipin Kumar. “Anomaly Detection: A Survey”. In: *ACM Comput. Surv.* 41.3 (July 2009), 15:1–15:58. ISSN: 0360-0300. DOI: 10.1145/1541880.1541882. URL: <http://doi.acm.org/10.1145/1541880.1541882>.
- [11] Lei Clifton et al. “Identification of Patient Deterioration in Vital-Sign Data using One-Class Support Vector Machines.” In: Jan. 2011, pp. 125–131.
- [12] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. New York, NY, USA: Wiley-Interscience, 2006. ISBN: 0471241954.
- [13] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. “Adversarial Feature Learning”. In: *CoRR* abs/1605.09782 (2017).
- [14] Vincent Dumoulin et al. “Adversarially Learned Inference”. In: *CoRR* abs/1606.00704 (2017).
- [15] Gilberto Fernandes et al. “Network anomaly detection using IP flows with Principal Component Analysis and Ant Colony Optimization”. In: *Journal of Network and Computer Applications* 64 (2016), pp. 1 –11. ISSN: 1084-8045. DOI: <https://doi.org/10.1016/j.jnca.2015.11.024>. URL: <http://www.sciencedirect.com/science/article/pii/S1084804516000618>.
- [16] Ugo Fiore et al. “Using Generative Adversarial Networks for Improving Classification Effectiveness in Credit Card Fraud Detection”. In: *Information Sciences* (Dec. 2017). DOI: 10.1016/j.ins.2017.12.030.
- [17] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. “Deep Sparse Rectifier Neural Networks”. In: *AISTATS*. 2011.
- [18] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016.
- [19] Ian J. Goodfellow et al. “Generative Adversarial Nets”. In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. NIPS’14. Montreal, Canada: MIT Press, 2014, pp. 2672–2680. URL: <http://dl.acm.org/citation.cfm?id=2969033.2969125>.
- [20] Ishaan Gulrajani et al. “Improved Training of Wasserstein GANs”. In: *NIPS*. 2017.

Bibliography

- [21] Phillip Isola et al. “Image-to-Image Translation with Conditional Adversarial Networks”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 5967–5976.
- [22] J. Jabez and B. Muthukumar. “Intrusion Detection System (IDS): Anomaly Detection Using Outlier Detection Approach”. In: *Procedia Computer Science* 48 (2015). International Conference on Computer, Communication and Convergence (ICCC 2015), pp. 338 –346. ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2015.04.191>. URL: <http://www.sciencedirect.com/science/article/pii/S1877050915007000>.
- [23] Dongil Kim et al. “Machine Learning-based Novelty Detection for Faulty Wafer Detection in Semiconductor Manufacturing”. In: *Expert Syst. Appl.* 39.4 (Mar. 2012), pp. 4075–4083. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2011.09.088. URL: <http://dx.doi.org/10.1016/j.eswa.2011.09.088>.
- [24] Taeksoo Kim et al. “Learning to Discover Cross-Domain Relations with Generative Adversarial Networks”. In: *ICML*. 2017.
- [25] Walter Kramer. “Probability & Measure : Patrick Billingsley (1995): (3rd ed.). New York : Wiley, ISBN 0-471-0071-02, pp 593, [pound sign] 49.95”. In: *Computational Statistics & Data Analysis* 20.6 (1995), pp. 702–703. URL: <https://ideas.repec.org/a/eee/csdana/v20y1995i6p703-702.html>.
- [26] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. “CIFAR-10 (Canadian Institute for Advanced Research)”. In: (). URL: <http://www.cs.toronto.edu/~kriz/cifar.html>.
- [27] Anders Boesen Lindbo Larsen et al. “Autoencoding beyond pixels using a learned similarity metric”. In: *ICML*. 2016.
- [28] Yann LeCun et al. “A tutorial on energy-based learning”. In: *PREDICTING STRUCTURED DATA*. MIT Press, 2006.
- [29] Chunyuan Li et al. “Towards Understanding Adversarial Learning for Joint Distribution Matching”. In: *NIPS*. 2017.
- [30] Markos Markou and Sameer Singh. “Novelty Detection: A Review&Mdash;Part 2: Neural Network Based Approaches”. In: *Signal Process.* 83.12 (Dec. 2003), pp. 2499–2521. ISSN: 0165-1684. DOI: 10.1016/j.sigpro.2003.07.019. URL: <http://dx.doi.org/10.1016/j.sigpro.2003.07.019>.

BIBLIOGRAPHY

- [31] Luis Martí et al. “Anomaly Detection Based on Sensor Data in Petroleum Industry Applications”. In: *Sensors* 15.2 (2015), pp. 2774–2797. ISSN: 1424-8220. DOI: 10.3390/s150202774. URL: <http://www.mdpi.com/1424-8220/15/2/2774>.
- [32] Takeru Miyato et al. “Spectral Normalization for Generative Adversarial Networks”. In: Feb. 2018.
- [33] Paolo Napoletano, Flavio Piccoli, and Raimondo Schettini. “Anomaly Detection in Nanofibrous Materials by CNN-Based Self-Similarity”. In: *Sensors* 18.2 (2018), p. 209. ISSN: 1424-8220. DOI: 10.3390/s18010209. URL: <http://dx.doi.org/10.3390/s18010209>.
- [34] Yuval Netzer et al. “Reading Digits in Natural Images with Unsupervised Feature Learning”. In: 2011.
- [35] Marco A. F. Pimentel et al. “Review: A Review of Novelty Detection”. In: *Signal Process.* 99 (June 2014), pp. 215–249. ISSN: 0165-1684. DOI: 10.1016/j.sigpro.2013.12.026. URL: <http://dx.doi.org/10.1016/j.sigpro.2013.12.026>.
- [36] Alec Radford, Luke Metz, and Soumith Chintala. “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”. In: *CoRR* abs/1511.06434 (2016).
- [37] Bengt Ringnér. “Law of the unconscious statistician”. In: (2009).
- [38] George G Roussas. *An introduction to measure-theoretic probability*. Burlington, MA: Elsevier, 2004. URL: <http://cds.cern.ch/record/1614120>.
- [39] Ruslan Salakhutdinov and Geoffrey Hinton. “Deep Boltzmann Machines”. In: *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*. Ed. by David van Dyk and Max Welling. Vol. 5. Proceedings of Machine Learning Research. Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR, 2009, pp. 448–455. URL: <http://proceedings.mlr.press/v5/salakhutdinov09a.html>.
- [40] Tim Salimans et al. “Improved Techniques for Training GANs”. In: (June 2016).
- [41] Thomas Schlegl et al. “f-AnoGAN: Fast Unsupervised Anomaly Detection with Generative Adversarial Networks”. In: *Medical Image Analysis* 54.Data Mining and Knowledge Discovery 29 3 2015 (2019), pp. 30–44. DOI: 10.1016/j.media.2019.01.010. URL: <https://app.dimensions.ai/details/publication/pub.1111824956>.

Bibliography

- [42] Thomas Schlegl et al. “Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery”. In: *IPMI*. 2017.
- [43] H.H. Sohrab. *Basic Real Analysis*. Birkhäuser Boston, 2011. ISBN: 9780817682323. URL: <https://books.google.it/books?id=zjfaBwAAQBAJ>.
- [44] Nitish Srivastava et al. “Dropout: a simple way to prevent neural networks from overfitting”. In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958.
- [45] Raihan Ul Islam, Mohammad Shahadat Hossain, and Karl Andersson. “A novel anomaly detection algorithm for sensor data under uncertainty”. In: *Soft Computing* 22.5 (2018), pp. 1623–1639. ISSN: 1433-7479. DOI: 10.1007/s00500-016-2425-2. URL: <https://doi.org/10.1007/s00500-016-2425-2>.
- [46] Bernard Widrow and Eugene Walach. “Appendix G: Thirty Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation”. In: 2008.
- [47] Zili Yi et al. “DualGAN: Unsupervised Dual Learning for Image-to-Image Translation”. In: *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 2868–2876.
- [48] Houssam Zenati et al. “Adversarially Learned Anomaly Detection”. In: *CoRR* abs/1812.02288 (2018). arXiv: 1812.02288. URL: <http://arxiv.org/abs/1812.02288>.
- [49] Junbo Jake Zhao, Michaël Mathieu, and Yann LeCun. “Energy-based Generative Adversarial Network”. In: *CoRR* abs/1609.03126 (2016).
- [50] Jun-Yan Zhu et al. “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks”. In: *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 2242–2251.