WASSERSTEIN GENERATIVE ADVERSARIAL ACTIVE LEARNING FOR ANOMALY DETECTION WITH GRADIENT PENALTY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HASAN ALI DURAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2021

Approval of the thesis:

WASSERSTEIN GENERATIVE ADVERSARIAL ACTIVE LEARNING FOR ANOMALY DETECTION WITH GRADIENT PENALTY

submitted by **HASAN ALI DURAN** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** _____

Assoc. Prof. Dr. Şeyda Ertekin
Supervisor, **Computer Engineering, METU** _____

Examining Committee Members:

Prof. Dr. Pınar Karagöz
Computer Engineering, METU _____

Assoc. Prof. Dr. Şeyda Ertekin
Computer Engineering, METU _____

Prof. Dr. Suat Özdemir
Computer Engineering, Hacettepe University _____

Date: 08.09.2021

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Hasan Ali Duran

Signature :

ABSTRACT

WASSERSTEIN GENERATIVE ADVERSARIAL ACTIVE LEARNING FOR ANOMALY DETECTION WITH GRADIENT PENALTY

Duran, Hasan Ali
M.S., Department of Computer Engineering
Supervisor: Assoc. Prof. Dr. Şeyda Ertekin

September 2021, 80 pages

Anomaly detection has become a very important topic with the advancing machine learning techniques and is used in many different application areas. In this study, we approach differently than the anomaly detection methods performed on standard generative models and describe anomaly detection as a binary classification problem. However, in order to train a highly accurate classifier model, the number of anomaly data in data-sets is very limited, and with synthetic data produced using generative models, it can be brought to a usable level to train the model. In our model like GANs while Generator produces potential informative anomaly data, the Discriminator tries to determine whether the generated data is fake or real. In addition to these, we have added the Critic network to our model in order to enable the Generator to produce more realistic and informative data. In this way, we designed our model the Discriminator to be trained with the data produced by the Generator which is improved by the Critic network. Therefore, after sufficient training, the Discriminator turns into a natural anomaly detection classification tool. Since the Generator produce more realistic data in each epoch during the training phase, created ones more informative potential anomaly data for the Discriminator, which will allow the algorithm to develop

with more informative data with active learning logic. Our novelty is a generative adversarial active learning (GAAL) structure designed over the Wasserstein Generative Adversarial Network with Gradient Penalty (WGAN-GP) instead of just applying this method over the standard GAN model. Both our Generator model can produce more realistic and more informative data than before, and at the same time, it prevents the mode collapse problem, which is one of the biggest problems of the standard GAN model. We have obtained a model that can detect anomalies with higher accuracy. Improved version of Wasserstein Generative Adversarial Active Learning (WGAAL-GP) has been tested on different data sets and the results obtained are presented by comparing them with previous studies.

Keywords: Deep Learning, Machine Learning, Generative Models, GAN, Wasserstein Gan, Outlier Detection, Anomaly Detection, Active Learning

ÖZ

GRADYAN CEZALI WASSERSTEIN ÜRETİCİ ÇEKİŞMELİ AĞLAR İLE AKTİF ÖĞRENME KULLANILARAK ANOMALİ TESPİTİ

Duran, Hasan Ali
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Şeyda Ertekin

Eylül 2021 , 80 sayfa

Gelişen makine öğrenmesi teknikleri ile anomali tespiti çok önemli bir konu haline gelmiş ve birçok farklı uygulama alanında kullanılmaktadır. Bu çalışmada, standart üretici modeller üzerinde gerçekleştirilen anomali tespit yöntemlerinden farklı olarak yaklaşılıyor ve anormallik tespitini bir ikili sınıflandırma problemi olarak tanımlıyoruz. Ancak, yüksek doğrulukta bir sınıflandırıcı modeli eğitmek için veri setlerindeki anomali verisi sayısı çok sınırlıdır ve üretici modeller kullanılarak üretilen sentetik veriler modeli eğitmek için kullanılabilir bir düzeye getirilebilir. GAN'lardaki gibi modelimizde Generator potansiyel bilgilendirici anomali verileri üretirken, Discriminator üretilen verilerin sahte mi gerçek mi olduğunu belirlemeye çalışır. Bunlara ek olarak Generator'ın daha gerçekçi ve bilgilendirici veriler üretebilmesi için modelimize Critic ağını da ekledik. Bu şekilde, Critic tarafından eğitilen Generator'ın ürettiği verilerle eğitilecek Discriminator modelimizi tasarladık. Bu nedenle yeterli eğitimden sonra Discriminator doğal bir anomali tespit sınıflandırma aracına dönüşür. Generator eğitim aşamasında her turda daha gerçekçi veriler üreteceğinden, Discriminator için daha bilgilendirici potansiyel anomali verileri üretecek ve bu da algoritma-

nın aktif öğrenme mantığı ile daha bilgilendirici verilerle gelişmesini sağlayacaktır. Çalışmamızda, bu yöntemi standart GAN modeli üzerinden çözmek yerine, Wasserstein Generative Adversarial Network'ün gradient cezasıyla (WGAN-GP) geliştirilmiş versiyonu üzerinden tasarlanmış üretken bir çekişmeli aktif öğrenme (GAAL) yapısıdır. Bu sayede hem Generator modelimiz eskisinden daha gerçekçi ve daha bilgilendirici veriler üretebilmekte hem de standart GAN modelinin en büyük sorunlarından biri olan mod collapse sorununun önüne geçmektedir. Bu sayede anomalileri daha yüksek doğrulukla tespit edebilen bir model elde etmiş olduk. Wasserstein Generative Adversarial Active Learning'in (WGAAL-GP) geliştirilmiş versiyonu farklı veri setleri üzerinde denenmiş ve elde edilen sonuçlar önceki çalışmalarla karşılaştırılarak bu çalışmada sunulmuştur.

Anahtar Kelimeler: Derin Öğrenme, Makine Öğrenmesi, Generative Modeller, GAN, Wasserstein Gan, Anomali Tespiti, Aktif Öğrenme, Outlier tespiti

To my lovely parents

ACKNOWLEDGMENTS

First and foremost, I would like to express my appreciation to my advisor, Assoc. Prof. Dr. Şeyda Ertekin, without whom my thesis would not have been possible. Her thoughts and point of view constantly pushed me ahead. It was a joy to collaborate with her.

I'd also want to express my gratitude to my lovely family for their unconditional support not just throughout the writing of this thesis, but also throughout my life. They have consistently encouraged me to achieve my aims.

I would also like to thank Tuğçe Alara Yılmaz, who has always supported me in this process and always been by my side, I know and want to be with me in this process and always.Without her support, this thesis would not have come to this day, I am very grateful to her for always taking me further when I had difficulties. I would like to thank her for sharing this support not only in thesis but also in every moment of life.

My very dear friend Yasin Berk Gültekin, with whom I always shared the most difficult times during the education process and overcame together, was one of my biggest supporters in this process as well. I will always be grateful for being so helpful in completing this thesis and for being with me throughout this process.

I would like to thank Osman Taşdelen for his knowledge about data and studies during the thesis process, his attitude that never hesitates to share this knowledge with me, and for his unconditional support in my studies.

I would also like to thank my colleagues for supporting me with their ideas and experiences during this process. Their experiences have been very helpful in getting me out of difficult situations.

And finally, I would like to thank all my friends for supporting me in this process and for being with me in the existence of the thesis. Without them, this thesis would not

be at this level, nor would I be this person.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

2D	2 Dimensional
3D	3 Dimensional
GAN	Generative Adversarial Network
WGAN	Wasserstein Generative Adversarial Network
WGAN-GP	Wasserstein Generative Adversarial Network with Gradient Penalty
GAAL	Generative Adversarial Active Learning
WGAAL	Wasserstein Generative Adversarial Active Learning
WGAAL-GP	Wasserstein Generative Adversarial Active Learning with Gradient Penalty
PCA	Principal Component Analysis
SVM	Support Vector Machine
OCSVM	One Class Support Vector Machine
KNN	K Nearest Neighbour
AGPO	Artificially Generating Potential Outliers
SOGAAL	Single Objective Generative Adversarial Active Learning
MOGAAL	Multiple Objective Generative Adversarial Active Learning
DCGAN	Deep Convolutional Generative Adversarial Network
KL	Kullback–Leibler
JS	Jensen-Shannon
EM	Earth Mover
G	Generator
D	Discriminator
GP	Gradient Penalty

AI	Artificial Intelligence
ML	Machine Learning

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

Anomaly, in the most basic terms, refers to something that does not fit into an expected situation or flow. It can also be described as abnormal or outlier in the literature. Anomaly detection, on the other hand, is the case of a specific data deviating too much from standard data within a data set, more simply, it is the detection of points that we can define as abnormal in the data. [67]
The process of discovering patterns in data that do not conform to a model of typical behavior is known as anomaly detection [68]. The ideas underpinning anomaly detection may appear complicated and unapproachable unless you're a data scientist or practitioner familiar with technologies that offer algorithms for pattern identification. However, the advantages are obvious.
The purpose of anomaly detection is to find cases that are out of the ordinary among data that appears to be similar[69]. Anomaly detection is a useful tool for detecting fraud, network intrusion, and other unusual events that may be relevant but are difficult to notice.

We should first understand why anomaly detection is an important problem. We can explain the importance of anomaly detection with the example below. For example, you, as a customer, spend around 250-300 dollars per month with your credit card. In other words, you are a customer in your bank's low-budget customer portfolio. But when you suddenly spend $1500 the next month, this transaction is a move away from low-budget customer portfolio behavior. If the bank notices this situation and informs you about this expenditure in some way, the customer will be able to notice

if he has been defrauded. Therefore, anomaly detection plays a very important role in this example or while examining MRI results in the medical field [60][61][62][66], we can distinguish the results of a normal person and a sick person, perhaps without the need for a doctor, or we will not overlook a problem that the doctor overlooked with advanced computers for anomaly detection. So anomaly detection tools has wide usage on various applications. [2],[7],[9],[10],[38],[13],[24]

Another area where anomaly detection application is important can be considered as aviation. In our study, we can understand the flight patterns and characteristics of passenger and cargo planes with the flight data of civil aircraft, for which we used a sample data. In this way, when flight movements that do not comply with the patterns of civil flights are noticed, it can be realized that it is not a normal flight. This awareness can prevent a possible danger, for example, a plane can be detected early or unmanned aerial vehicles can be prevented from flying in areas where there is no permission. We can show this as a direct use in the field of air defense in the defense industry and military systems.

1.2 Proposed Methods and Models

Machine Learning is used in anomaly detection approaches. Machine learning can be used to learn a system's properties from observed data, which can help to improve detection speed. Machine-learning algorithms are capable of not only learning from data, but also making predictions based on that data, as well as improving their predictive abilities by "learning" from the outcomes of their initial predictions as events unfold in real life (the feedback loop). Anomaly detection using machine learning encompasses approaches for detecting and classifying abnormalities in vast and complicated big data sets. Sequential hypothesis testing are one way for detecting anomalies. For identifying changes in the distributions of real-time data and setting alarm settings, such as cumulative sum charts and sequential probability ratio tests can be used.

Recently, a wide variety of machine learning methods have been used for anomaly detection. Basically, there are 4 different main approaches to anomaly detection process.

First, statistical methods [1],[4],[6] secondly linear based methods [2],[5],[7],[14] thirdly distance based methods [8],[9],[12],[21] and finally reconstruction-based [16],[27],[22],[24] methods. In the statistical method, which is the simplest method, the similarity of the incoming data with the general average and its standard deviation are calculated and its difference from the mean is calculated. By comparing the result with a threshold value, it is decided whether there is an anomaly or not. As another method, the most popular distance based method is K nearest neighbor [8],[9],[12],[21]. The most popular linear based method is Principal component analysis [15],[16],[17]. The method that we can describe as the most advanced is reconstruction based methods. Among these, one of the most popular methods used for anomaly detection is Auto-encoder [16],[17],[22], but the method used in our study is not one of them.

However, in our study, we developed an anomaly detection tool with generative adversarial networks which is one of the most popular reconstruction based deep learning methods of recent times. Generative adversarial networks (GAN) [18] have become much more useful and popular with recent studies[24],[27],[29], [28]. After that, it is possible to use it as an anomaly detection tool. First of all, it is necessary to understand the working principle of GAN structure in general. GAN is basically a model consisting of two different networks, a model in which these two networks (generator - discriminator) play the min-max game mutually. Networks improve each other thanks to this mutually contentious work. GAN is used as an anomaly detection tool in two different ways. The first [29] and the more popular one is to make an anomaly decision by evaluating how close the data is created by using the generator's reconstruction capabilities. The second method that works with active learning logic in GAN has many example studies [30],[31],[3]. In this method, it determines the anomaly by generating potential anomaly data using the generator, enabling the discriminator to make the distinction between fake data and real data much more clearly. Thus, it evaluates the anomaly detection as a classification problem and uses the discriminator as a classifier. Our method is also a method that uses the GAN method for anomalies by generating potential anomaly data. In the GAAL with anomaly detection study [30], anomaly detection is made using the standard GAN structure.

1.3 Contributions and Novelties

As an additional novelty to GAAL based anomaly detection study [30], we produced more informative potential anomaly data by supporting this method with Wasserstein GAN with gradient penalty [25],[26], not with standard GAN. In this way, we both produced more informative data and prevented the mode collapse problem. In our study, the Wasserstein GAN with gradient penalty (WGAN-GP) cannot be used alone because the discriminator working in the WGAN-GP model does not discriminate directly on fake data or real data. In this study, which renames the Discriminator network as Critic. Critic measures the distance of the generated data to the real data, so it does not act directly as a classifier. Therefore, we need a classifier and we build our model on three different networks. Two networks (Generator and Critic) come via WGAN-GP, in addition to these, we also set up a Discriminator network. In this context, while the generator network improves itself with the feedback it receives from the Critic, both the Critic and the Discriminator develop itself with the data and feedback produced by the Generator. As a result, we call this model Improved version of Wasserstein Generative Adversarial Active Learning (WGAAL-GP). During the Generator training process, each epoch will improve itself and will be able to produce more informative data. The model combines GAN with the active learning context as it tries to produce anomaly data close to reality instead of producing data very similar to normal data directly. Thus, the discriminator network, as a classifier, can draw its boundaries much more accurately with more informative data and perform anomaly prediction and detection with higher accuracy.

As a summary our contributions can be summed up as follows:

- Compared to other studies, Wasserstein Gan is used in this study, so it produces more realistic potential anomaly data, so that while the algorithm is being trained, it is trained with sufficient and more realistic anomaly data. In this way, it prevents the problem of low number of anomaly data under normal conditions.

- By using Wasserstein Gan with gradient penalty, it prevents the mode collapse problem that will occur during the training process due to the nature of this

algorithm. In this way, more serious problems in making wrong decisions that may occur during testing or decision-making are prevented. In addition, it is ensured that the resulting accuracy scores are at a higher level.

- By using Wasserstein Gan and the standard Gan Discriminator jointly, the potential anomaly data produced in each epoch is produced more realistically, ensuring that the data produced in each epoch is more informative. In this way, it increases the level of informativeness by improving the active learning logic compared to other GAAL studies.

1.4 The Outline of the Thesis

The organization of the thesis can be expressed as, Chapter 2 researches and the chapter with the necessary background to understand our solution proposal. This chapter summarizes the techniques used in our proposed solution on 5 different topics. Chapter 3, on the other hand, is the chapter where the previous studies on the subject of anomaly detection are summarized and a general literature search is available. In this chapter, which is divided into 2 main titles, first anomaly types are explained, then different algorithms used for anomaly detection are explained. In Chapter 4, we describe what kind of methodology we follow and how our proposed solution works. Chapter 5 is the section where we present our experiments on different datasets and our comparison of their results. This section is evaluated under two main headings, and in the first section, the introduction of the datasets and specifying their properties are made, while the second section includes the tests and interpretation of these tests. In our last chapter, Chapter 6, we make a summary conclusion of our study and present our suggestions that can be done as future work.

CHAPTER 2

BACKGROUND

2.1 Anomaly Detection

The technique of detecting outliers in a data-set is known as anomaly detection. Outliers are data items that stand out from the rest of the data-set and do not follow the expected pattern of behavior. Anomaly detection techniques have a wide range of applications [2],[7],[9],[10],[38],[13],[24] in business, science, and security, where isolating and acting on outlier detection findings is crucial. Algorithms such as categorization, regression, and clustering can be used to find abnormalities. Any of the supervised data science methods may be used to discover anomalies if the training data-set contains items with known anomalous outcomes.There are specific (unsupervised) algorithms that identify outliers without the need of a labeled training data-set, in addition to supervised techniques. In the case of unsupervised anomaly detection, algorithms can estimate distance from other data points or density in the data point's vicinity. Another method for identifying abnormalities is to rebuild data and compare reconstruction losses. Even clustering algorithms may be used to discover anomalies. Because the outlier is so far away from the other data points, it generally forms a distinct cluster from the others.

To summarize, using supervised and unsupervised techniques, it is called anomaly detection to detect data that is farther or more irrelevant than it should be from the normal data or the general distribution of the data. Among the currently used anomaly detection methods[37] [38] [39] [57] [64], the most popular ones among the unsupervised ones can be considered as reconstruction based algorithms. Since these algorithms use deep learning techniques more efficiently, they allow better results than traditional methods. Although anomaly detection in these methods is different in

general (supervised or unsupervised), it somehow detects the behavior and distribution of the normal data and then tests the suitability of the data for this distribution or behavior. If the values obtained as a result of this test are more than a predetermined threshold value, the data is considered an anomaly. If this result is lower than the threshold value, it is considered normal like other data. Basically, all algorithms work on this logic, the differences with each other are the methods and capacity to understand the behavior of normal data. At the same time, the ways of comparing new data change according to the normal learning method in the past during the testing phase. For example, in a distance-based method [34], the distance of the new point to the nearest n points is calculated and averaged. This average distance taken is compared with the predetermined threshold value and it is decided whether it is an anomaly or normal data [21]. However, in auto encoders [22], which is a reconstruction-based method, a neural network is obtained by training the auto encoder model with normal data. This neural network first compresses the data by reducing its size, then expands the compressed state and brings it back to data size. The weights of the model are trained according to the normal data. Afterwards, when an abnormal data comes in the testing phase, this data is taken as an input to this model, which has been trained with normal data before. Then, the reconstruction loss of this reconstructed data is calculated and this loss value is compared with the previously determined threshold value. If it is higher than the threshold value, it is decided that there is an anomaly, if it is lower than the threshold value, it is considered a normal data. As a result, although the methods of the algorithms change, as the general logic, the method of calculating the difference between the new data and the normal data changes, but the anomaly is detected by measuring the difference.

2.2 Deep Learning

Machine Learning, on the other hand, is a subset of Artificial Intelligence, while Deep Learning is a subset of Machine Learning. Artificial intelligence (AI) is a broad phrase that refers to methods that allow computers to emulate human behavior. All of this is made possible through machine learning, which is a set of algorithms based on data. Deep Learning, on the other hand, is a form of Machine Learning that is inspired

by the human brain's structure. Deep learning algorithms analyze data with a predetermined logical framework in order to reach similar conclusions as humans. Deep learning does this by employing a multi-layered structure of algorithms known as neural networks.The neural network's architecture is inspired by the structure of the human brain. Neural networks can be trained to do the same tasks on data that our brains do when identifying patterns[40] and classifying different sorts of information.Also Deep learning can be used for prediction, many studies [41][42][43][44][45] prove that.

Individual layers of neural networks may also be thought of as a kind of filter that works from the most obvious to the most subtle, improving the chance of detecting and producing a right result.The human brain operates in a similar manner. When we acquire new knowledge, our brain attempts to compare it to previously encountered items. Deep neural networks make use of the same principle. We can use neural networks to accomplish a different tasks, such as grouping, regression, and classification. We can use neural networks to group or sort unlabeled data based on similarities between the samples. Alternatively, in the instance of classification, we can train the network on a labeled dataset in order to categorize the samples in the dataset.Deep learning models can solve challenges that basic machine learning models can't do efficiently. Artificial neural networks offer unique characteristics that allow deep learning models to handle tasks that basic machine learning models can't do efficiently.

As a result, to summarize deep learning, it is a technique that can do feature extraction on its own without the need for preprocessing such as machine learning, in cases where machine learning cannot do it or works more unsuccessfully, so it is a technique that tries to make sense of data very similar to how the human brain works. Models built on this similarity are also called neural networks. If the working logic is to be explained in a simple way, these neural networks are composed of different numbers of nodes. And a network is formed in such a way that these nodes are completely dependent or limited to each other. The properties of the data are transferred to the model from the nodes on the input layer, then this data starts to be processed in the hidden layers, if any, and as a result of the process, it is transferred to the next layer. Finally, since we have designed our output layer to consist of as many nodes as we

want, the size of the output is at the level we want. While reaching the output layer, the final operation is done on the incoming data and the output value is reached. Well, to talk about what is the process done in each layer, it can be summarized in the simplest form as follows. There is a bond created with a certain weight between the nodes connected to each other, then these weights are updated to approach the desired output at each step of the training process, allowing the model to give more accurate results. The update process is provided by calculating gradient descent by taking the derivative of the process there. Thanks to this gradient descent, it is decided how the weight should be updated and it is increased or decreased accordingly in the simplest way. During the training process, the result for each input and the required result are compared, and then a difference is calculated. Then, using this difference, gradient descents are calculated and the weights of the connections between the nodes are updated. As a result, when the training process is over, this model, whose weights have been created before, gives a new test data to this model as an input and the result it reaches in the output layer is the desired result. We can summarize deep learning in the simplest way like this. We can obtain larger models by using different models in combination with each other.

2.3 Generative Adversarial Networks

Goodfellow et al. in 2014 [18] have introduced a new generative unsupervised learning neural network. GAN's are specifically trained to learn a mapping G: z $\rightarrow$ X to data distribution P_{real} on which they are trained, so that the network models and approximates the distribution P_{model} and is able to produce new data points that fall in the data distribution P_{model}. A probabilistic generative model called generative adversarial networks consists of two neural networks, a Generator, and a Discriminator that compete in a min-max game. The Discriminator takes a data point X from X_{real} and X_{fake} as an input and predicts whether it is real (from training data) or generated by a Generator (G(z)), where G(z) has a label of 0 and X has a label of 1. Generator G, on the other hand, takes a random noise vector z from a preset latent space z $\rightarrow$ Z as an input and generates a synthetic data point G(z) that the Discriminator recognizes as coming from the real data distribution P_{real}. The high level overview is illustrated in

the figure 2.1. Because these two networks are connected, during the training both are improving in their tasks through back-propagation. The Generator becomes better at producing synthetic data in a way that the Discriminator gets fooled into predicting it as real, while the Discriminator gets better at distinguishing real from synthetic data.

During the training, a Generator and a Discriminator play the min max game with a value function V (G, D), which is an objective of the GAN, as formulated in equation 2.1:

$$\mathbf{Min}_G\mathbf{Max}_D V(G,D) = \mathbf{E}_x[\log(D(x))] + \mathbf{E}_z[1 - D(\log(G(x)))] \tag{2.1}$$

GANs, or Generative Adversarial Networks, are a type of generative modeling that use deep learning techniques such as convolutional neural networks.In machine learning, generative modeling is an unsupervised learning job that entails automatically identifying [52] and learning regularities[53] or patterns[54] in input data such that the model may be used to create or output new instances that might have been drawn from the original data-set. GAN's are a clever approach to train a generative model by identifying the problem with two sub-models as a supervised learning issue: the model that we train to generate new models and the model that tries to categorize instances as real (from the domain) or as false (generated). In zero sum games, the two models will be trained jointly in adversarial ways until about half the time is tricked, meaning that the model Generator creates believable instances. GANs, in essence, generate their own training data. The Generator will begin to produce higher-quality output as the feedback loop between the adversarial networks continues, and the Discriminator will grow better at detecting falsely manufactured data.

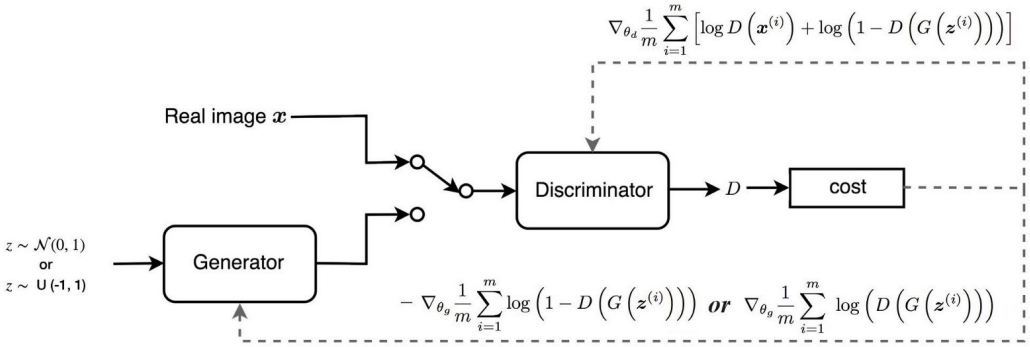

Figure 2.1: GAN's Structure [35]

Defining the intended end result and gathering an initial training data-set based on those parameters is the first step in constructing a GAN. This data is then randomized and fed into the Generator until it achieves basic output accuracy.

The produced pictures, together with real data points from the original concept, are then put into the Discriminator. The Discriminator sorts through the data and assigns a probability between 0 and 1 to each image's genuineness (1 correlates with real and 0 correlates with fake). The success of these values is manually checked, and the process is repeated until the desired result is achieved.

GANs are an exciting and quickly evolving field that promises to create generative models that are capable of creating realistic examples across a range of problem fields, particularly in image-to-image translation, for example in translating pictures in summer to winter and day after night.

2.3.1 Evolution of Generative Models

There is no common approach to evaluating and comparing generative models, which is an ongoing research area. The likelihood is the default metric because the GAN's objective function is a binary cross-entropy. The main concept is to create a function that is proportional to the quality of the photographs generated. Based on the estimation of the distribution of the real image embeddings in the Discriminator, the approach constructs the likelihood function [63]. Then, using the embeddings of the Discriminator, this function can determine the likelihood that an image corresponds

to the learned data distribution P_{data}. Models can, however, have a high probability but suffer from issues such as mode collapse (section 2.3.3).

However, there are a few alternative options. For example, in [32], they conducted an experiment with human annotators in which they were asked to compare created and actual photographs and try to discern between them. However, when human annotators are weary or unmotivated, they are more likely to make mistakes. Furthermore, if the photographs contain fuzzy or confusing objects or patterns, this strategy becomes more difficult for humans.

The Inception score is an option provided in [32]. The conditional label distribution P(y | G(z)) is calculated using the pre-trained Inception neural network model on the generated images. This label distribution P(y | G(z)) should have low entropy if the image contains meaningful items (so that the Inception model is more certain about what it identifies in it) (so small uncertainty). We wish to have a variety of generated images, in addition to photographs with relevant objects. That is, the label distribution among different generated samples must have a high entropy. Researchers in [32] introduce metric by integrating these two criteria as follows:

$$exp(\mathbf{E}_x KL(p(y|x)||p(y))) \tag{2.2}$$

2.3.2 Instability of Training GAN's

GAN training is a difficult and insecure procedure with no systematic approach to solving the problem. There are several reasons for this, the most important of which is that the Generator and Discriminator are trained individually without a unified loss function, and one of them might be over-trained in comparison to the other. Another problem is the lack of a sufficient metric for evaluation. Because we don't know anything about how well the network is operating based on the actual loss. Low loss of a Generator, for example, can indicate either that the Generator is good or that the Discriminator is terrible, resulting in poor Generator training while fooling the Discriminator with bad data. Furthermore, there is no method to identify when the learning procedure should be stopped or to compare different models. However, there

are a number of approaches that can help to stabilize the training:

- Normalizing inputs between -1 and 1
- Loss function can be modified. $\mathbf{E}_z[1 - D(\log(G(x)))]$ term of GAN equation should be updated $\mathbf{E}_z[D(\log(G(x)))]$
- Rather than using a uniform distribution, sample a noise vector z from the normal distribution.
- Separate batches of real and fake data were used to train the Discriminator.
- If the Discriminator or Generator's loss becomes too small, stop training them.
- Add noise to the Discriminator's inputs and/or to the Generator's layers with Gaussian noise.
- In the Generator, use dropout (50 percent).
- Using Adam optimizer [36]

2.3.3 Mode Collapse

The effect of mode collapse occurs when the Generator produces the same or nearly the same output for any given noise vector from latent space z. This occurs because the Generator learns to produce a single output that is constantly deceiving to the Discriminator. When the output is different but only by a slight fraction, it is called partial mode collapse. Normally, we would want as many different outputs as feasible. However, it is frequently noticed that data diversity and quality are sometimes mutually exclusive. The addition of label smoothing is one technique proposed in [30]. That is, instead of using the labels l = 0 and l = 1, try using the new labels l = 0.1 and l = 0.9.

Typically, you'll want your GAN to generate a wide range of outputs. You might want a distinct face for each random input to your face Generator, for example If a Generator provides a very believable result, however, the Generator may learn to exclusively produce that output. In fact, the Generator is continually striving to discover the one

output that the Discriminator thinks is the most believable.The Discriminator's best technique is to learn to always reject the same output (or a small set of outputs) if the Generator starts producing the same output (or a small set of outputs) over and over again. However, if the following generation of Discriminator becomes stuck in a local minimum and fails to identify the optimum strategy, the next Generator iteration will find the most feasible output for the present Discriminator far too easily.

Each iteration of the Generator over-optimizes for a specific Discriminator, and the Discriminator never learns how to escape the trap. As a result, the Generators alternate between a limited number of output kinds. Mode collapse is the term for this type of GAN failure.

2.3.4 Wasserstein Generative Adversarial Networks

Wasserstein generative adversarial network (WGAN) is a form of GAN introduced in a publication by M. Arjovsky et al. [25] in 2017 that addressed the challenges of training stability and loss function interpretability. Furthermore, it has been discovered that WGANs are extremely vulnerable to mode collapse.

The Generator Equation for GAN:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} log(D(G(z^{(i)}))) \tag{2.3}$$

The Generator Equation for WGAN:

$$\nabla_{\theta} \frac{1}{m} \sum_{i=1}^{m} f(G(z^{(i)})) \tag{2.4}$$

The Discriminator Equation for GAN:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [logD(x^{(i)}) + log(1 - D(G(z^{(i)})))] \tag{2.5}$$

The Critic Equation for WGAN:

$$\nabla_{\omega} \frac{1}{m} \sum_{i=1}^{m} [f(x^{(i)}) - f(G(z^{(i)}))] \tag{2.6}$$

The goal of GANs is to minimize f-divergence (the distance between two distributions = P_{model} and P_{data}) over region D in order to learn the distribution of the data P_{data}). In the min max game, this convergence in classical GAN is understood as minimizing Jensen Shannon (JS) or KL divergence. The authors of the publication [25] discuss the shortcomings of such metrics and propose that Earths Mover's/Wasserstein distance be used instead. In a physical world metaphor, this distance metric could be expressed as determining how much work is required to transfer one distribution to another, which is equal to the amount (mass) multiplied by the distance traveled. To put it another way, it seeks to find the smallest/easiest route to get from one pile to another (learned distribution P_{model} to the real distribution P_{data}). The authors of the paper show that there are some distributions for which KL or JS divergence fails to find a solution but EM distance does.

The following are the major architectural differences in the WGAN:

- The output of the Discriminator is no longer sigmoid and no longer represents the probability.
- The difference between the generated and real samples is the Discriminator loss.
- The Discriminator is trained n times more than the Generator (n = 5 according to the paper).
- Weight clipping: Using 1-Lipschitz in the Discriminator weights to force them to have small values centered around zero.

The research shown that utilizing Earth's mover's distance as a candidate for the divergence of the two distributions is a strong candidate. Weight clipping, on the other hand, causes the weights to be pushed in one direction, and clipping them can sometimes result in loops. This issue was addressed in [26] by introducing a more advanced WGAN.

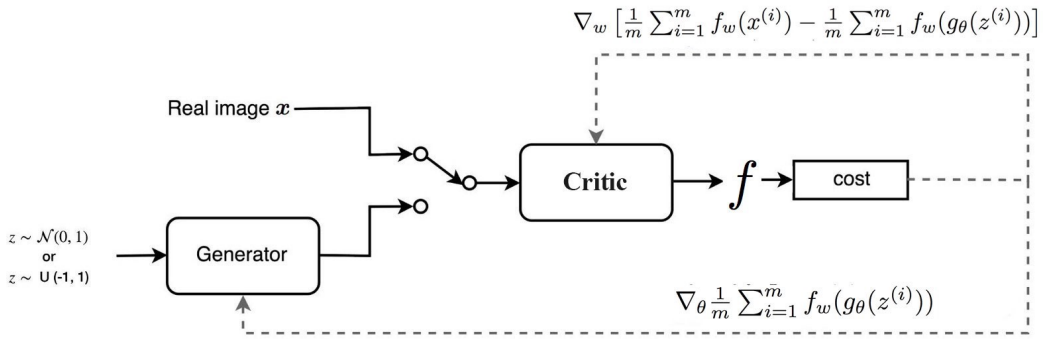

Figure 2.2: WGAN's Structure [35]

2.3.5 Wasserstein Generative Adversarial Networks with Gradient Penalty

The Wasserstein generative adversarial network with the gradient penalty [26] (also known as upgraded WGAN) solves the problem of weight clipping in the Discriminator by removing it and replacing it with the gradient penalty. Instead of clipping the weights, it penalizes the Discriminator's gradient with respect to the input. $L_{total} = L_{real} + L_{fake} + L_{GP}$ is the total loss function of an upgraded GAN. Gradient penalization occurs by itself, as discussed in [17], because it is included in the general loss function. As a result, the authors advise against using batch normalization. For the critic (Discriminator in WGAN), batch normalization is avoided . Correlations between samples in the same batch are created via batch normalization. Experiments show that it has an effect on the gradient penalty's efficiency. Some new cost functions, whether by design or not, include a gradient penalty in the cost function. Some of it is based only on empirical evidence that models misbehave as the gradient rises. Gradient penalty, on the other hand, increases computational complexity that may or may not be desirable, but it does result in higher-quality data.

For past implementations such as vanilla GANs, DCGANs, and WGANs, Batch Normalization was utilized to stabilize the training process. Batch Normalization, on the other hand, has an impact on the mapping of the input to the output since it causes association between samples from the same batch. Because WGAN-GP penalizes the norm of the critic's gradient for each of its inputs independently, it replaces Batch Normalization with Layer Normalization.

2.4 Active Learning

To train with decent results, most supervised machine learning models require a substantial amount of data. Even if this remark appears simplistic, most firms struggle to offer this data, particularly tagged data, to their data scientists. The latter is necessary for every supervised model to be trained and can constitute a major bottleneck for any data team. Most of the time, data scientists are given large, unlabeled data sets and asked to train high-performing models on them. In most cases, the amount of data is too large to manually categorize, making it difficult for data teams to train good supervised models with it.

The technique of prioritizing the data that has to be labeled in order to have the most influence on training a supervised model is named as Active learning [70]. Active learning can be utilized in instances when there is too much data to label and it is necessary to prioritize labeling the data in a wise way.[72]

The main hypothesis in active learning is that if an algorithm can choose the data from which it wishes to learn, it is more effective than standard methods with significantly fewer training data [71]. These are jobs that entail acquiring a huge amount of data that has been randomly sampled from the underlying distribution and using that data to train a model that can make a prediction. This is what you'd call a normal approach of passive learning.

The collection of labelled data is one of the most time consuming tasks in passive learning. There may be limiting factors in many settings which impede the collection of large volumes of labelled data. Take the pancreatic cancer study as an example. You may want to predict if a patient has pancreatic cancer, but you could only have the opportunity to perform a few more exams to gather features, etc. In this case, we can choose patients based on certain criteria rather than randomly selecting them. One example could be the drinking of alcohol over forty years in the patient dataset. This criteria must not be static, but may change according to previous patients' results. This might be your new criteria if, for example, you realize that your model is good at predicting pancreatic cancer for those over 50 years of life, but strive to predict exactly for people over 40-50 years. The selection process of those patients (or more generally instances) is called active learning based on the data we have gathered to date.

2.4.1 Steps for Active Learning

In the literature we are examining several approaches to how data points are prioritized in labeling and how they are iterated. Even though, only the most common and simple methods will be presented. The steps to use active learning on an unlabelled data set are:

- The first thing that has to be done is to properly label an extremely small sample of this data.
- Once a small number of labeled data is available, the model needs to be trained. The model will of course not be great, but will help us understand which parameter space areas must first be labeled to improve it.
- The model is used, once the model is trained, to predict the class of every unlabeled data point that remains.
- A score is selected based on the model prediction on each unlabeled data point. There may be various score methods.
- If the best approach to prioritization of the labeling is chosen, it can be repeated iteratively: A new model can be training on a new, label-based data set. Upon training the new model with the data subset, the unlabelled data points can be used to update priority scores to continue labelling. Thus the labeling strategy can be optimized as the models improve and improve.

2.5 Generative Adversarial Active Learning

Generative adversarial active learning is a technique [31] that enables smarter selection of inputs in the training process, which is the basic working principle of active learning, by using generative adversarial networks. Before explaining this selection process directly, it is necessary to examine how a standard selection process works in active learning under normal conditions. There is a sample pool with training data. In each epoch, a certain number of samples are selected from this pool. This selection process should be chosen among the data that is closest to the boundaries of the

classifier and the model is uncertain, which can be done with different techniques in the literature. After this selection process takes place, this data needs to be labeled. It can be said that the main purpose of active learning is to reduce this labeling cost. Because instead of labeling all of the data, it allows this operation to be performed only for a specially selected subset of the model that has been determined to be most useful for the model. In this way, it is possible to save a lot of development level and labeling cost without sacrificing the self-development of the model.

However, in order for these useful data to be selected over the existing data set, there must be sufficient informative data. This is not possible for every dataset because datasets that are unevenly distributed or the number of data is very limited can be used. Not every dataset is optimal and very usable. In these cases, training the model is not easy even with standard active learning. This is where Generative Adversarial Networks come into play. It has been seen that Generative Adversarial Networks can basically learn from the training dataset and produce synthetic data, and this data is produced as high quality and similar to real data as possible. This capability of Generative Adversarial Networks is actually a very convenient way to expand limited datasets and to contribute more to the development of models. In this way, we will be able to obtain data in which the classifier will be much more uncertain. Even if this data is synthetic, it is very close to the truth, so it will not cause a problem in terms of similarity to the dataset, which is not actually in our dataset.

These artificial data produced allow much more data to be selected for the active learning algorithm [31]. As the number of this data increases, it turns into a much more instructive algorithm compared to the classical algorithm. Due to the nature of the GAN, the informativeness level of the data increases, as it creates more realistic data in each epoch. Basically, in the classical algorithm, the data that will develop the model were chosen wisely, but the selection set was only as much as the training sample pool, so the development of the model was limited to a certain level. However, when Generative Adversarial Networks are included in the event, this data set is not only limited to the sample pool, but this pool is supported by artificial data produced by the Generator, and since the informative level of this data is very high, it greatly increases the development of the model. Generative adversarial active learning can be summarized in this way.

CHAPTER 3

LITERATURE SEARCH

3.1 Types of Anomalies

Anomaly detection is basically a method that is eliminated in 3 different ways. Point anomalies, contextual anomalies and collective anomalies.

3.1.1 Point Anomalies

Point anomalies, contextual anomalies and collective anomalies. The part called point anomalies is simply a type of anomaly that occurs when a single specific point is distant from other normal points in space in which the data is located. However, in order to define this, the data must be in a format that can be defined in certain spaces and the distance can be measured.

3.1.2 Contextual Anomalies

Contextual anomalies as the second type of anomaly. Contextual anomalies can be basically defined as follows, if a data is defined as an anomaly in some cases and as normal in some cases, that is, if the anomaly is determined according to the context in which it is located and is evaluated as an anomaly, this type of anomaly is called contextual anomalies. To illustrate this situation, for example, 3 degrees weather is considered a normal air temperature in winter, if the weather is measured as 3 degrees any day in July in the summer season, we consider this situation as an anomaly. This is an example of contextual anomalies.

3.1.3 Collective Anomalies

The last type of anomaly is Collective anomalies. In the case of collective anomalies, some data alone do not indicate an anomaly situation, but when the data is combined and the whole dataset is examined, we can infer that this situation is actually an anomaly. To illustrate this situation, a person's heart rate range can range from 60 to 150. If the 60 relaxing time pulse value , 150 can be considered as the exercise time heart rate value. However, if the pulse rate of 1000 people rises to 150 and above at 11 o'clock at night, this actually indicates an anomaly. Because it is unlikely that 1000 people exercise in the middle of the night, this situation may indicate a decrease in oxygen level in the environment. Therefore, to summarize, collective anomalies is a type of anomaly that occurs when data that do not appear as an anomaly alone point to an anomaly when combined.

3.2 Anomaly Detection Methods

Anomaly detection techniques can be handled in many different ways and thus classified in many different ways.There are plenty of works done before [37] [38] [39] [57] [64]. These classifications can be made according to algorithms or techniques. One of the most comprehensive classifications is shown in the figure 3.1.

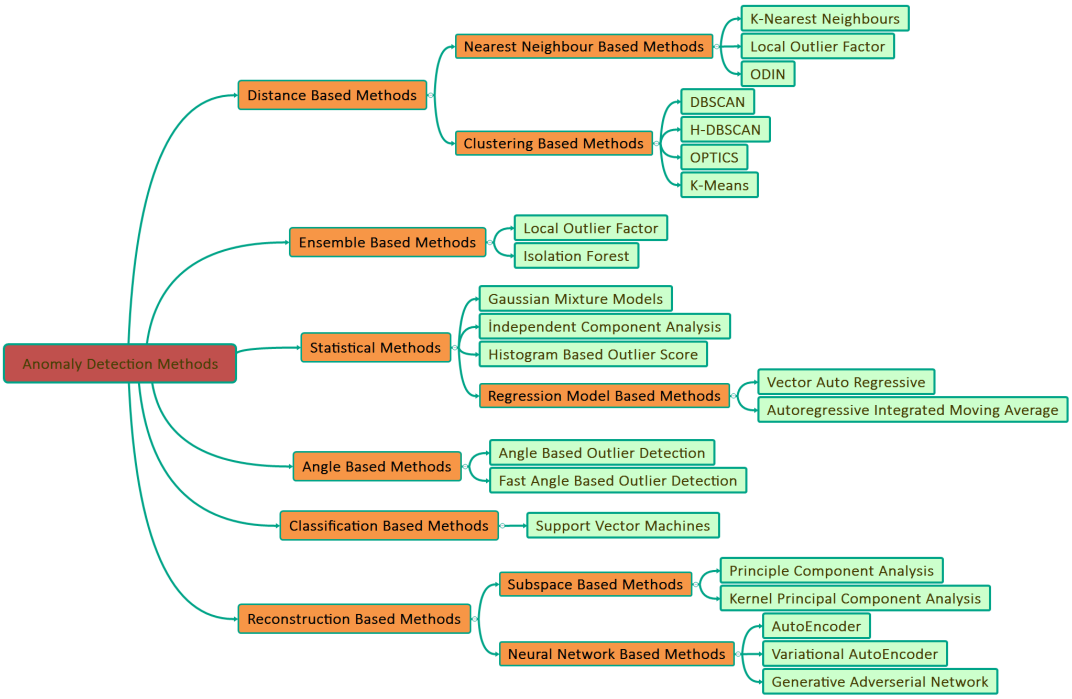

Figure 3.1: "Different Type of Anomaly Detection Techniques"

But going over all of the classifications here may not make much sense. With a more basic classification, the best practices of these methods and the best quality studies made on these methods should be evaluated. When developing models for anomaly detection, unsupervised methods are used rather than supervised methods due to the lack of sufficient number and level of labeled data. The unsupervised methods used are basically divided into three. The first is linear-based methods, the second is distance-based methods, and finally, deep learning methods.

3.2.1 Linear-Based Methods

The most popular approach among linear-based methods is Principal Component Analysis (PCA). Principal component analysis is basically a method of calculating the projection of a multidimensional data in space to a lower dimensional space so that the maximum variance is preserved. The line with the lowest average distance to all points is determined from the multidimensional space of all points in the data. Then another line is determined perpendicular to the first line and this process can be

repeated as many times as desired. In general, a threshold is determined for this process to converge, this threshold is determined according to the variance value. When it falls below the threshold value, the process should be terminated, otherwise, this process may not benefit. The lines obtained as a result are defined in a way that they form the foundations of the space that the projection of the data will create. And the data is updated so that all data are expressed here.

So how is this method used as an anomaly detection tool? PCA based anomaly detection study [20] can be summed up as follows.The data that needs to be decided whether there is an anomaly or not is converted into a designed space with new dimensions previously calculated. Then, the distance of this projected point to all lines of space in reduced dimensions is calculated and the average distance score is obtained. If this score is above the previously determined threshold, this actually means that the data given in the space with reduced dimension number is expressed farther from the original data. This situation is interpreted as that data is an anomaly.

However, this method has different problems. First of all, in order for this method to give correct and healthy results, the training data used should consist of highly correlated points. If this situation is not provided, the fundamental lines in the reduced space cannot be selected with good quality. In addition, the data used is expected to be in the Gaussian distribution format, otherwise the data will be very scattered and there will be no healthy representation in the reduced space. As a result, Principal Component Analysis may not always be a very good method for anomaly detection due to the problems mentioned above.

3.2.2 Distance Based Methods

Among the Distance-based methods, the most popular approach is K-Nearest Neighbor [34]. The nearest neighbors algorithm is considered one of the most basic machine learning algorithms. The basic working principle of the algorithm can be thought as follows; First of all, a value of k is determined and all subsequent steps will proceed in relation to this k value. When a new individual data arrives, the distances to the old points are calculated. In general, the distance measurement method here is Euclidean distance, but Manhattan distance and Minkovski distance methods

can also be used. Then, from the calculated distances, the nearest k points to the new point are found. The weighted average is taken according to the distances of this k point. In this averaging process, the farther should have a lower weight, mathematically it can be thought of as a $\frac{1}{d}$ ratio. Then, the location where the new point will be found from the space of the model as a result of this proportioning is calculated and added to the space as a new point. When the next point is reached, this process is repeated, and all previously placed points should be checked, and the operation of the algorithm should continue. Due to the natural functioning of the algorithm, the K nearest neighbors algorithm is an algorithm that does not require training. The testing and training process can be thought of as jointly occurring.

If we need to evaluate the K nearest neighbors algorithm as an anomaly tool [21], we can explain what differences it exhibits as follows: again, we need to find the position of the new point as if we apply the standard operation of the algorithm, and while applying this process, we need to find a total distance by taking the weighted average of k nearest neighbors. Using these distances, an anomaly score is calculated in any way we want (the simplest method is the addition process). Then we compare it with a threshold value that we previously defined, if this anomaly score is higher than our threshold, we can evaluate the new incoming data as an anomaly.

But KNN is an algorithm that starts to slow down very quickly as the data-set grows. Also, similar to this situation, if the data has a multivariate structure, it becomes increasingly difficult to determine the location of the new point. Also, if the data expresses different features in different units, it becomes difficult to use this algorithm. If the features of the data are in different ranges, the distance calculation algorithms cannot give accurate results. All the features should be in the same scale if you want to calculate the effects and distances correctly. It is necessary to work on such data. In addition, it is necessary to determine the value of k as optimum, if it is determined as 1, it may be overfit, if it is determined as very large values, it may not give Discriminatory results. In addition, one of the biggest problems of this method is that it has problems with imbalanced data, if there is an outlier, this algorithm is very sensitive in the training set, which affects the algorithm and prevents other results from being healthy. As a result, this method is not always seen as an efficient algorithm for anomaly detection due to the reasons mentioned above.

3.2.3 Deep Learning based Methods

Finally, the most popular anomaly detection methods used today are deep learning based methods[55].The most popular two algorithm of the deep learning based anomaly detection methods is Auto-encoders and Generative Adversarial Networks.

3.2.3.1 Auto-encoder based Anomaly Detection

Auto-encoders are one of the most frequently used of these methods. Auto-encoder method can be described as one of the most basic methods of deep learning algorithms. In fact, when considered simple, the auto-encoder is a compression and rebuilding mechanism that consists of two parts. Its two basic parts are defined as encoder and decoder. Encoder tries to bring the given data from the multidimensional expression method to the data size on the latent space, and it tries to understand only the important features of the data in less dimensions and compresses it in this way. Decoder, on the other hand, tries to generate data in the size of real data from the vector expressed as compressed in latent space. The dimensions of the data produced by the decoder as a result and the data input to the encoder must be equal. The layer in the latent space is called hidden layer. The main purpose is to use the hidden layer as a bottleneck and force it to extract important features of the data. When defining the model, if the hidden layer layer is defined as n layers instead of a single layer, this model is called a deep auto-encoder. In the training phase, the reconstructed data and the original form of the data are compared and a loss value is calculated from the differences. This calculated loss value is called reconstruction loss.

Basically, when the method is examined, it can be considered as a dimension reduction method and it is very similar to the Principal Component Analysis method in this regard. But the main difference between them is that Principal Component Analysis is a linear dimension reduction method, while Auto-encoders are a non-linear method of reducing the dimensions of the data. If the activation function for the Auto-encoder is defined as linear, this method actually converges to Principal Component Analysis.

If we evaluate auto-encoders as an anomaly detection tool [22], in fact, a very different technique is not used from other methods. Encoder and Decoder models come out

as sufficiently advanced models after sufficient training. After a new data arrives, it is first encoded and then this data is given as input to the decoder from the latent space and reconstructed. A reconstruction loss is calculated between the resulting data and the original data. It is then compared with a predetermined threshold value. If this loss value is higher than the threshold value, this data is considered an anomaly.

Auto-encoders are actually a very data dependent method, even if they are the most capable of the algorithms mentioned earlier. If the training data cannot express the world to be tested sufficiently, it may not give very efficient results. Also, auto-encoders need a lot of data to be trained. During the training phase, the model tries to express the data in less dimensions, but tries to preserve all its features as much as possible, but does not give information about which feature is more important or not. For these reasons, although it is superior to other algorithms for anomaly detection, there may be situations where it is not efficient.

3.2.3.2 Generative Adversarial Network based Methods

Another deep learning algorithm that has been used recently is Generative Adversarial Networks. Before using it as an anomaly detection tool, it is necessary to talk about what a generative adversarial network is. [18] The generative adversarial network is actually a structure based on two different models trying to fool each other in a way that they mutually provide developer output. It consists of two different models in basic structure. The name of one is defined as Generator and the name of the other is defined as Discriminator. Generator develops as a model that tries to produce data very similar to real data. While doing this, it uses a random latent vector basis to produce data in the size of the real data and the values it should be. It tries to learn the distribution of real data during the training process, and a model is formed that can produce data suitable for the distribution of real data as a result of adequate training in optimum condition.

At the same time, the Discriminator takes on a different task. Discriminator takes a piece of data in the dimensions of the real data as input, and the provider of this input is undertaken by both the real training data and the fake data produced by the Generator. Discriminator's task is to understand whether the source of the incoming

data is real training data or a fake data originating from a Generator.

However, in this case, the improvement of one model means that the other model is actually weak because the strong model becomes able to deceive the other. In fact, this situation constitutes the main idea of mutual development and development that underlies the general idea of GAN. The two models mutually contribute to each other's development as if they play a minimum maximum game with each other. After the training phase is completed, we have two different models, one of which is Generator: a model where we can produce random data very close to the real data, Discriminator: a model that can detect the existence of this difference and perceive the fact that when data is different from the real data. This method can be regarded as the state of the art in creating new data in a very realistic way. GAN methodologies used for many different applications it is proven many studies[46][47][48][49][50][51]. However we focus on anomaly detection studies and tasks.

It has proven to be an important tool for anomaly detection, even if it has previously been used mainly to generate new pictures or to transfer data in different domains[56][58] [65] to each other.Also literature has different studies [59] of GAN's as anomaly detector. First of all, it is stated in this study [23] how the GAN model will turn into an anomaly detection tool. The system here is first designed to generate mutual input. Generator is trying to produce a copy of the original data using a random latent vector. Discriminator takes both the data produced by the Generator and the real data as input and tries to predict which data is real data and which data is artificial data. In the training process, the Discriminator and Generator develop each other mutually. After sufficient training, if the Discriminator detects a data as fake , this data can be evaluated as an anomaly. Likewise, when the distance between a data produced by a sufficiently trained Generator and the suspicious data is calculated, if this distance is above a certain threshold, this data can also be evaluated as an anomaly. After the two networks have turned into separate anomaly detection tools, we have been able to use these two together or separately for anomaly detection. In this study, two networks are used separately, but a total score is calculated and it is decided whether there is an anomaly or not. Coefficient values are determined separately for Discriminator and Generator, the sum of these two values must be 1. Then the values provided by the two networks are multiplied by coefficients. The two

results we obtained are added together and the value we have obtained in total can be considered as our common anomaly score. Anomaly is determined by comparing this score with our Threshold values. As a result, two networks can produce separate anomaly scores, in this way, a total anomaly score is obtained and GAN can be used as an anomaly tool.

One of the other fundamental studies on this topic is Anomaly Detection with Generative Adversarial Networks for Multivariate Time Series [24]. In order to explain the related study, it is necessary to explain the Long Short Term Memory mechanism first. When the data is evaluated as time series, historical data can affect the data in the present time. Therefore, it is necessary to predict future data or evaluate the past while producing new data. Multiple methods are used to evaluate historical data in the machine learning world. Recurrent Neural Network and Convolutional Neural Network mechanisms can be given as examples. However, the study shown as state of the art on this subject is the Long Short Term Memory method. Basically, while designing the model, each node of the model is designed dependent on the previous node and a data sharing is provided between them. In general, when sharing this data, the sigmoid function is chosen as the activation function. But Relu and Leaky Relu functions can also be selected. The logic of the activation function is as follows, the result of the activation function varies between 0 and 1, if it is 1, it ensures that the data passes completely, if it is 0, it prevents the transition completely. With the activation function, nodes connected to each other transmit data from the past to the current node, and since the newest of the past nodes transmits data with the highest coefficient and the ones in the past with a much less coefficient, the dependence on the past is set up correctly. How many nodes were interested in the past can be set parametrically and this data is called window size. As a result, LSTM method can be integrated into networks and is a method used to obtain efficiency in time series data by evaluating historical data.

In this study, instead of using a standard Generator and Discriminator network, a Long Short Term Memory based network was built. These networks are then organized to be used as Generator and Discriminator. In this way, the data can be evaluated as a time series instead of being meaningful alone. Only the residual loss and reconstruction loss that occur on the Generator side become available efficiently for anomaly

detection. The model is developed using these obtained loss values. Anomaly detection is made by using the anomaly detection method applied in the standard GAN structure over the model that has reached its final form as a result of sufficient training. The biggest difference is that historical data is evaluated over LSTM while training, and in the same way, the data becomes meaningful with the historical data in the inference section and anomaly detection is made in this way. A single data is not taken as input, but historical data is evaluated as a total. Basically, the difference from the standard Gan Anomaly process can be defined as the LSTM based infrastructure.

In addition to these issues, there is a very common problem in the GAN method. The name of this problem is the Mode Collapse problem. In fact, it is basically the problem of Generator going on a single line and generating uniform data due to this, losing data diversity. A study to solve this problem is [25] [26] Wasserstein GAN and Wasserstein GAN with Gradient Penalty. In this study, unlike the standard GAN, the Wasserstein distance method is used in distance calculations. This method is basically based on finding the minimum number of moves one of the data sets containing two different distributions can be transformed into the other. In addition, Discriminator can be trained much faster, as a result, the Generator tries to move forward from the data piece it is strong and brings the mode collapse problem with it. In order to prevent this, gradient penalty is used to penalize situations where the data is not used completely, thus Generator mode can be blocked without falling into a collapse state. As a result of this study, the model can escape from the mode collapse problem compared to the standard GAN, and in addition to this, even if Discriminator completes its development, Generator can continue to learn. In addition, by using Wasserstein distance calculation, the situation such as not being able to converge is prevented. [33] In Haloui's study Wasserstein Gan is used for the anomaly detection

Another study on this subject is Efficient Gan Anomaly [27]. Basically, this study was built on the infrastructure of another study without turning it into an anomaly detection tool. The name of this work is Bidirectional Generative Adversarial Networks. This work, called BiGAN [28] for short, differs from the standard Generative Adversarial Networks. This difference basically occurs with the inclusion of another network in the model, while Generator and Discriminator continue to develop each other mutually, a new Encoder network is included in the environment. Generator

tries to generate fake data from a random latent space vector, while the Encoder plays the opposite role of Generator. Trying to generate latent space vectors from real dataset data. In addition to this work, the role of Discriminator needs to be updated. Discriminator not only takes fake or real data as input, it also takes latent space vectors as input. Thus, the clusters to be decided by the Discriminator are defined as follows; Real data with fake latent space vector or fake data with real latent space vector. When the training process is over, we have a Generator network that can produce realistic data and an encoder network that can express a real data in latent space.

There is no difference in the training part in the "Efficient Gan Anomaly" study, which creates an anomaly detection tool using the BiGAN method. In the inference part, it compresses the incoming test data using the advanced Encoder network in hand, then the latent space vector obtained becomes an input for the Generator, and the Generator network generates a new data using this vector, the obtained data is compared with the original test data and the difference is calculated. In addition, the Discriminator network turns into a natural anomaly detection tool like in other studies. A total anomaly score is obtained by multiplying the values obtained by these two methods with the determined coefficients. If the calculated anomaly score is above the determined threshold value, the data is considered as an anomaly.

Another study among the anomaly detection studies conducted with the Generative Adversarial Network is the study named GAN-omaly [29]. It was used in Autoencoders as an extra for Generator and Discriminator. In this way, the author tried to get the good features of two different deep learning algorithms, and a more complex network model was obtained by integrating the two algorithms. This model basically includes one encoder, a decoder and a Generator consisting of an encoder, and a Discriminator. The first encoder (encoder, which is part of the Generator) encodes the incoming data and the resulting latent space vector enters the decoder part of the Generator as input. This network outputs a data in the size of the original data, this data is used as input to two different networks, the first is the Discriminator network and the second is the other encoder network. The second encoder network converts the data back to the latent space vector and a loss is calculated by comparing it with the first latent space vector, and it improves itself by using the corresponding loss

value. Generator and Discriminator improve itself in the same way as in other models.

When we evaluate this study as an anomaly detection tool, we first insert the data into the encoder part of the Generator and save the resulting latent space vector, then we input the relevant latent space vector to the decoder part of the Generator. We encode the result produced by Generator thanks to our other encoder network and obtain a second latent space vector. Then we compare the first latent space vector with the latent space vectors we have produced last. We compare the benchmark result with the threshold value we have determined before. If it is higher than our threshold value, we consider the data as an anomaly, if it is less than the threshold value, we understand that it is normal.

Another method solving anomaly detection problem as a binary classification problem, by generating artificial anomaly data, which also plays a major role in our study.[30] The difference of this method from the standard artificial data generating methods is that it can produce much more realistic and more qualified artificial data by performing this process with GAN. In fact, in this method, GAN networks take two different roles. Generator tries to produce anomaly data that is very close to the real data instead of trying to produce an exact replica of the data itself. Discriminator acts as a classifier trying to understand whether the incoming data is coming from real data or fake data. In this case, the Discriminator turns into a natural anomaly detection classifier. To train a classifier under normal circumstances, there must be sufficient data for each class in the training data. However, anomaly data are not frequently encountered data, so we increase the number of anomaly data by using a high-fidelity Generator. In this way, our classifier can improve itself much better. In addition, due to the natural working principle of GAN, Generator will start to produce more realistic anomaly data at every stage of training, and these data will turn into much more informative data for the Discriminator. In this way, it will be able to detect an anomaly by combining active learning logic with GAN.[30], [31]

CHAPTER 4

METHODOLOGY

In this section, we detail our methodology. In order to fully explain our methodology, let's first explain the working principle of generative adversarial active learning (GAAL) based anomaly detection study [30]. First, consider the anomaly detection problem as a binary classification problem. There are two different classes and these classes are labeled as anomaly data and normal data. The main purpose of the study is to determine a division boundary for these two different classes and to draw this division boundary as sharply and accurately as possible. In order for a classifier to create this boundary more accurately, it should be trained with sufficient data belonging to each class. Otherwise, the created boundary is quite rough and cannot be classified at the desired levels. However, as previous studies clearly show, there is a dramatic difference between normal data and the number of abnormal data in a standard data set. This has the potential to be a big problem when the anomaly detection problem is considered as a classification problem. In addition, since we do not have prior label information for the data, we must be able to detect anomaly data during the training process of the model. To solve this problem, all of the training data is evaluated as a uniform distribution and the data calculated at a distance above a specified threshold value is evaluated as anomaly. However, this situation becomes more difficult as the size of the data grows.Because it is more difficult to express high-dimensional data with a uniform distribution, there may be situations that do not converge, and it is much more difficult when making similarity measurements. Due to the insufficient number of anomaly data, the necessary information for drawing the division boundary cannot be learned sufficiently by the model. When these operations are not done correctly enough, the classifier can find results that are very different from what is desired. In order to solve this problem, it is necessary to support the model with

artificial anomaly data. Standard Artificially Generating Potential Outlier (AGPO) based methods are obtained by randomly generating the anomaly data in this training data. However, this amount may still be insufficient for high dimensional data like Figure 4.1. If we make the number of anomaly data as high as possible, eventually our classifier will get much sharper division boundary if we cover every place outside the Gaussian distribution space, which our data describes as normal, with potential anomaly data.

To do this, we have to generate an exponential number of potential anomaly data. The more informative this potential anomaly data is, the more accurate and sharp our classifier is. The GAAL for unsupervised anomaly detection [30] study solved this problem using generative adversarial networks. In our study, as a novelty, we generate the potential anomaly data with the Improved version of Wasserstein Generative Adversarial Network (WGAN-GP) [26] to be more informative, thus making the division boundaries of our classifier more accurate and sharp. To illustration of this algorithm can be seen in figure 4.2.

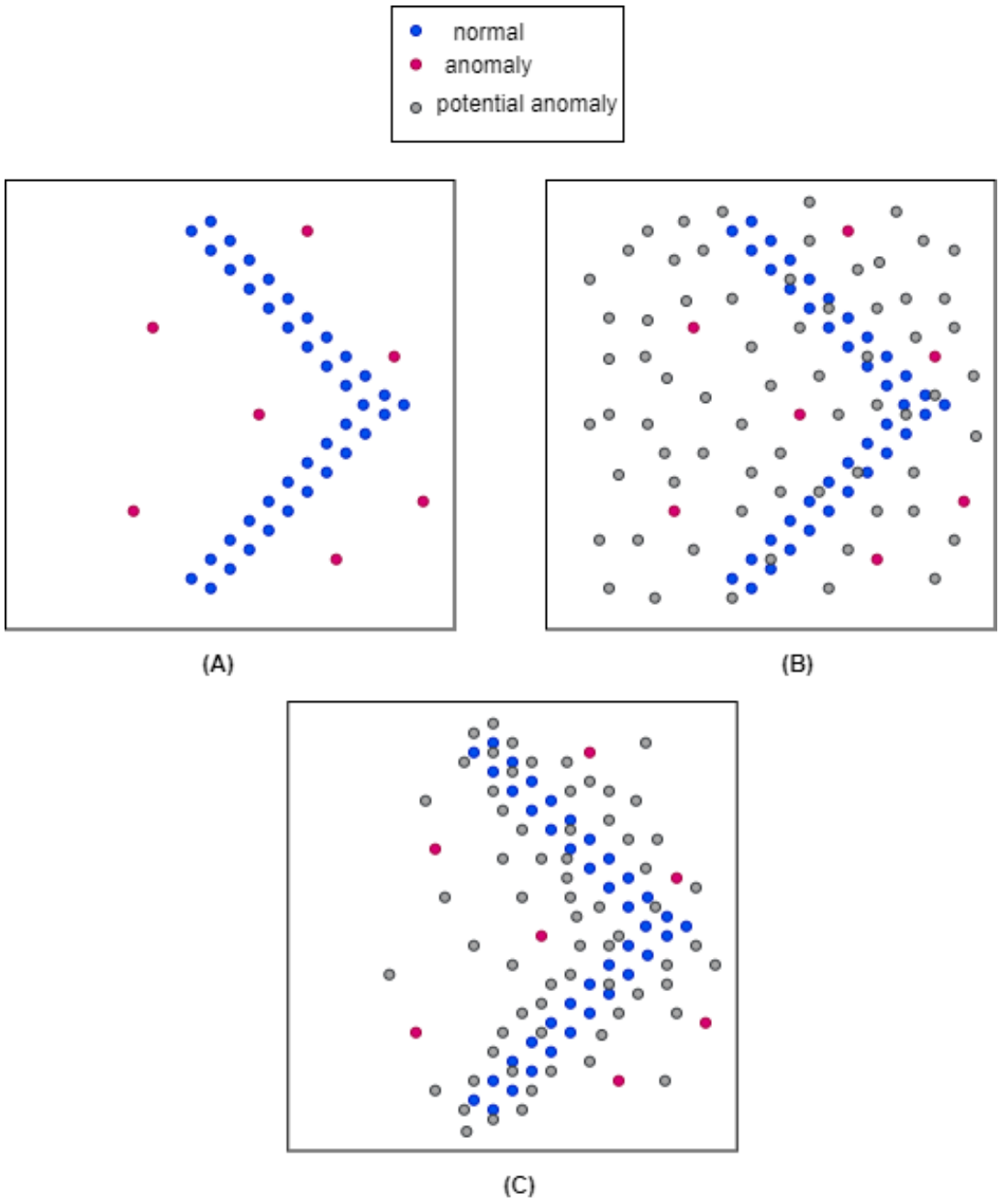

Figure 4.1: A illustrates real data, B illustrates AGPO based generated potential anomaly points, C illustrates GAAL based generated potential anomaly points

Before going into the details of this method, the working principles of Generative Adversarial Networks should be examined. Generative adversarial networks are one of the most popular deep learning algorithms of recent times. This method can be seen as two networks playing a min-max game mutually. When examined in more

detail, we can evaluate the generator network as a network that tries to produce real-like data by using a noise vector as an input. In this case, the Discriminator network is trying to understand by using the data as input, calculating the probability of whether the related data is generator or belongs to the real data set. G represents the Generator and D represents the Discriminator. Also x randomly selected data from real data and z represents the noise vector for the Generator. The GAN optimization formula can be summed up mathematically as like Equation 4.1.

$$\min_G \max_D V(D,G) = E_{x\sim P_{data}(x)}[\log D(x)] + E_{z\sim P_z(z)}[\log(1 - D(G(z)))] \quad (4.1)$$

In our study, similar to the GAAL study [30], the generator using the noise vector as an input was assigned to produce data with potential anomalies. Discriminator, on the other hand, distinguishes between the data produced by the generator and the real data, and in fact, it undertakes the task of creating a division boundary between anomaly data and normal data. In this case, we can qualify Discriminator as a classifier, and as a result of sufficient training, a Discriminator that has shown sufficient development turns into an anomaly data normal data classifier. In the early stages of training, our classifier cannot draw a very clear division boundary, as the generator cannot yet produce more realistic potential anomaly data. However, as the generator learns the distribution of real data much better in the later stages of the training process, it begins to produce much more realistic data. This realistic data produced becomes much more challenging for the classifier, which actually means that the data are much more informative. In this way, the classifier starts to draw its division boundary much more sharply and accurately and determines its own boundaries to cover normal data in the smallest volume. forcing. This process is basically realized by the fact that the data produced by the generator in each training step is more realistic. More realistic artificial data means more informative potential anomaly data, which means that we actually increase the level of information at each training step, which is the active learning process itself. These anomaly data, which are created more rationally than the standard artificial anomaly data generating methods (randomly generated ones), further enhance the capabilities of the classifier. We can obtain a much sharper division boundary between anomaly and normal by obtaining

more informative data by using the Generator's capacity to learn the distribution of data at a very high level. In addition, with this method, we can use the ability of the GAN model to learn the structure of the data strongly in high-dimensional data, and we can create an anomaly detection tool that is much less affected by the size of the data.

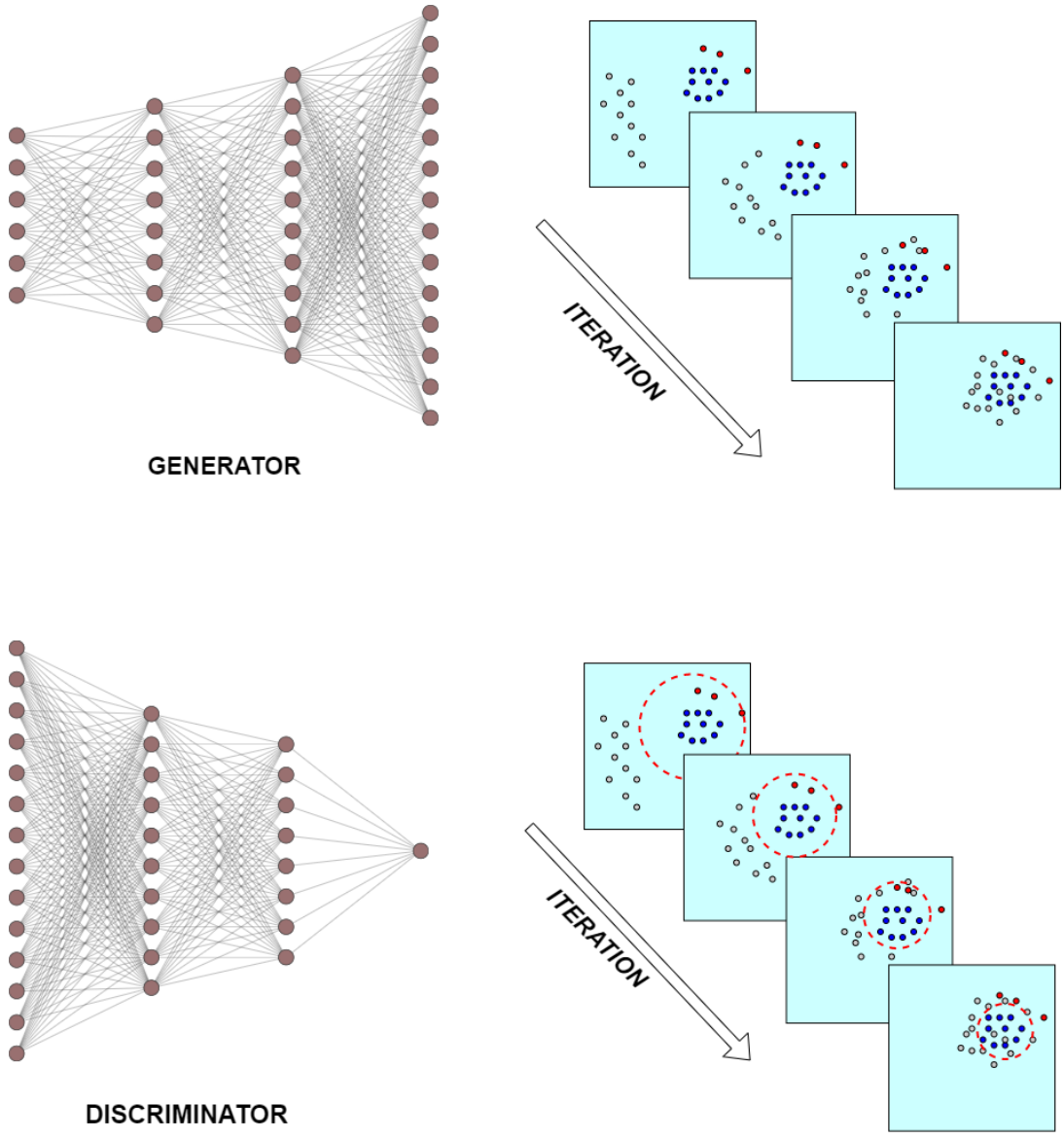

Figure 4.2: GAAL based anomaly detection algorithm. The red dots illustrates that anomaly points, the blue dots illustrates normal points, the grey ones illustrates potential anomaly points

But the biggest problem facing this study is the mode collapse problem. What is mode

collapse problem? If the generator starts producing the same output (or a small set of outputs) over and over again, the discriminator's best strategy is to learn to always reject that output. But if the next generation of discriminator gets stuck in a local minimum and doesn't find the best strategy, then it's too easy for the next generator iteration to find the most plausible output for the current discriminator. Each iteration of generator over-optimizes for a particular discriminator, and the discriminator never manages to learn its way out of the trap. As a result the generators rotate through a small set of output types. This form of GAN failure is called mode collapse. In order to solve this problem, a different technique was applied in the GAAL study [30]. Instead of generating potential anomaly data with a single Generator, they tried to prevent the mode collapse problem by using the number of generators as a parametric value and using more generator networks. However, in order to use this solution, it is necessary to choose the right number of generators and this number is directly related to how many clusters the data consists of. Since we do not have prior information about the data, it is difficult to decide on the number of generators and to guarantee that any of the selected generators will not encounter the mode collapse problem again. In this study, we build a similar structure to GAAL study [30], with Improved Version of Wasserstein Generative Adversarial Network (WGAN-GP) as a novel to solve this problem and aim to increase data realism and informativeness while avoiding the mode collapse problem. However, due to the natural structure of WGAN-GP, it is not enough to use WGAN instead of GAN while this process is taking place, we also have to modify the general structure accordingly.

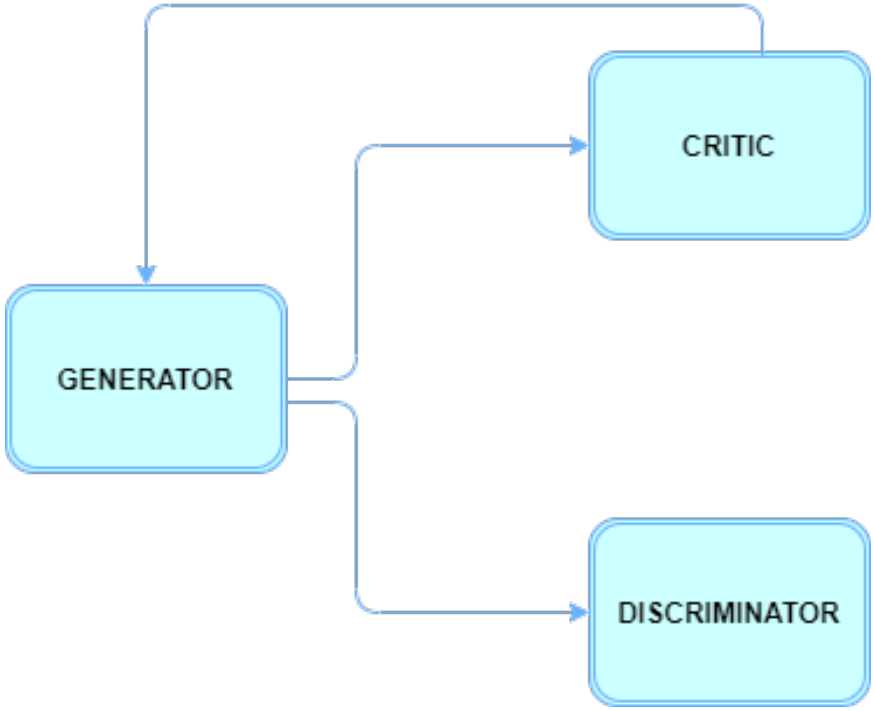

Figure 4.3: The feedback mechanism of our structure.

To explain the new structure in figure 4.4 is illustrated our general structure. This structure does not contain only a generator and a discriminator as in the standard GAAL structure. Because in the standard GAAL structure the discriminator is used as a classifier, The discriminator is actually assigned as a distinguish mechanism between fake data and real data. In the GAAL structure, this classifier actually works as a classifier between anomaly and normal data. In order to prevent the mode collapse problem, which is the most important problem of this structure, when we try to build a structure again with Wasserstein GAN version of GAAL, it is not enough to simply remove the standard GAN structure and replace it with Wasserstein GAN.

Because in the standard GAN structure, the cost function used to understand how realistic the data produced by the generator is is KL divergence or JS divergence. Let's call the distribution of real data P and let's call the estimated data distribution Q, also assume that the distributions of these data are Gaussian. When the difference between the estimated distribution and the distribution of the actual data is 0, these two data sets are considered to be the same. As the mean of Q increases, the divergence increases. The gradient of the divergency will eventually diminish. We have

close to a zero gradient, i.e. the generator learns nothing from the gradient descent. One of these cost function methods, KL divergence that mathematically illustrated in Equation 4.2, calculates it with a logarithmic-based formula and this method does not provide a symmetrical result. In order for this formula to give symmetrical results, this method has been updated by taking the average of two different distributions and calculating the sum of the distances of the distributions to this mean as mathematically illustrated in Equation 4.3, this method is also called JS divergence.

$$D_{KL}(P||Q) = \sum_{x=1}^{N} P(x) = \log \frac{P(x))}{Q(x)} \tag{4.2}$$

$$D_{JS}(P||Q) = \frac{1}{2} D_{KL}(P||\frac{P+Q}{2}) + (Q||\frac{P+Q}{2}) \tag{4.3}$$

WGAN solved the problem that the generator stopped developing due to the vanishing gradient problem experienced by the cost functions mentioned above by proposing a new cost function. The name of this proposed cost function is Earth-Moving Distance (Wasserstein Distance). This method basically calculates the similarity between distributions based on the cost of transforming one distribution into another. However, there may be an infinite method of converting one distribution to another. In this method, the cheapest conversion plan that will enable one distribution to transform into another is calculated. As a summary, the Wasserstein distance is the minimum cost of transporting mass in converting the data distribution q to the data distribution p as mathematically illustrated in Equation 4.4.

$$W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} E_{(x,y)\sim\gamma}[||x - y||] \tag{4.4}$$

With this proposed [25] cost function, the WGAN study made the distance between distributions measurable everywhere with a smoother gradient. In this way, the learning of the generator does not stop after a certain period of time and our model becomes able to produce higher quality data. But this Wasserstein distance is highly intractable as can be seen at Equation 4.5. This is why we need a 1-Lipschitz function to calculate least upper bound. In this case, we get a very similar structure to the discriminator

in our network, but here the results do not come from a sigmoid function, so we get a score result directly instead of obtaining a probability. Therefore, this structure turns into a network that generates a score about how realistic the data is. Along with his new role, his name is used as a critic instead of a discriminator. But as a deficiency in this critic network, f function has to be 1-Lipschitz. In order to provide this situation, as a very simple method, WGAN clips the weights in the range of values specified as hyper parameter. This clipping forces the model to provide Lipschitz constraint so that the Critic Wasserstein calculates the distance. However, in some cases, very realistic data cannot be obtained and the model is very dependent on hyper parameters to solve this problem instead of clipping [26] WGAN-GP study proposes a new technique, Lipschitz formula can be seen at Equation 4.6, a differentiable function f is 1-Lipschitz if and only if it has gradients with norm at most 1 everywhere. Therefore, the gradient norms should be around 1. Instead of applying clipping, the WGAN-GP penalized model if the gradient norm value moves away from the norm value 1. Thus, we obtain a model that produces higher quality data.

$$W(P_r, P_\Theta) = \sup_{||f||_L \leq 1} E_{x \sim P_r}[f(x)] - E_{x \sim P_\Theta}[f(x)] \tag{4.5}$$

$$L = (\underset{\widetilde{x} \sim P_\Theta}{E}[D(\widetilde{x})] - \underset{x \sim P_r}{E}[D(x)]) + (\lambda \underset{\hat{x} \sim P_{\hat{x}}}{E}[(||\Delta_{\hat{x}} D(\hat{x}\,||)_2 - 1)^2]) \tag{4.6}$$

We use these studies on our anomaly detection model as follows: With the new cost function proposed by the WGAN-GP study, the Critic network was included instead of a Discriminator, which is different from a standard GAN structure. With this difference, as mentioned above, the Critic network does not produce its results as a result of a sigmoid function, so it generates a scalar score and does not directly calculate probability. When we evaluate this situation together with the GAAL study, it is not possible to use it directly because the discriminator in the GAAL study [30] acts as a classifier as a task, but instead of acting as a classifier, the Critic network is only a score generator on how much the generated data is similar to the distribution of real data. We want to produce more realistic data and therefore more informative potential anomaly data, but we also want to approach the problem of anomaly detection as a classification problem and obtain a classifier thanks to the discriminator.

However, although WGAN-GP solves our first request, our second problem persists. In order to overcome this problem, as a novelty, we designed our model to consist of 3 different networks instead of only generator and discriminator. These networks are designed to be Generator, Discriminator and Critic. However, in the working mechanism here, we have slightly updated the development and feedback processes of networks, not all inter-network connections are two-way. First of all, as you can see from the Figure 4.3, we send the data generated by the generator as input to the other two networks. Both the discriminator and the critic also take the data of the actual data set as input. In this way, a generator plays a direct role in the development of two different networks with the data it produces. However, the generator does not receive direct feedback from the discriminator like the old GAAL structure during the development process, it evaluates the scores produced by the Critic network as feedback in order to develop the generator and updates its network accordingly. In this way, Generator can produce much more realistic data and these data become much more informative potential anomaly data. The discriminator that develops itself using this data and will turn into an anomaly-normal classifier as a result of adequate training. Similar to the old GAAL build, Discriminator develops itself with real data and fake data. And it draws a division boundary between these two data. However, since fake data is produced by supporting WGAN-GP method, it becomes more realistic and more informative potential anomaly data. In each phase of the training, more informative data will be given to the discriminator from the generator network, which tries to produce more informative data than the previous round using the feed-backs of the critical network. This situation is actually considered as a model that develops itself with the logic of active learning, as more informative data are always selected and produced as in the GAAL study. In this way, as a result of sufficient training, we get a Discriminator trained with much more informative data than before. Then we get this discriminator as an anomaly detection classifier with more reliable and high accuracy results. At the same time, we prevent the mode collapse problem as the generator network is trained with the WGAN-GP [26] method. Therefore, the multiple objective generative adversarial active learning [30] method used in the GAAL study, to summarize simply, does not need to use n generators instead of one. In this way, we both produce more informative potential data and get rid of the mode collapse problem. This results in a Discriminator network with sharper borders. A sharper division

boundary makes the discriminator in the classifier task work better while detecting more anomalies.

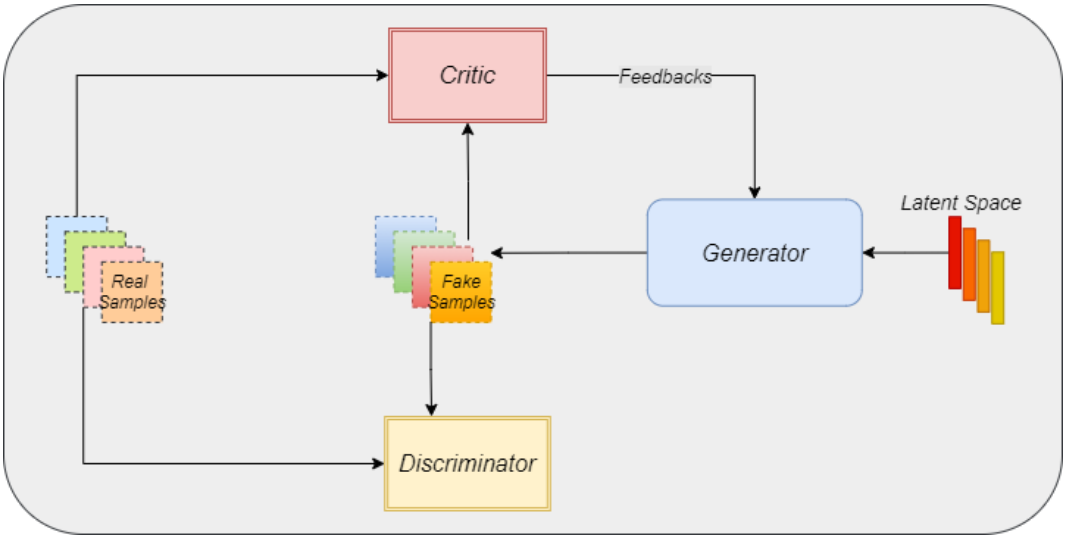

Figure 4.4: Our general structure.

If we want to summarize the logic and methodology of the study, we should first consider anomaly detection as a classification problem. Then, as in standard GAAL-based anomaly detection studies [30], the discriminator is developed as the first tool for anomaly detection in the later stages. Because the discriminator is basically a classifier. A classifier that distinguishes between artificially produced data and data that actually exists on the dataset. But standard generative adversarial networks have some problems. One of the most important of these problems is the mode collapse problem and the realism and quality level of the data. In order to overcome these problems, instead of using the standard generative adversarial network directly, we use the Wasserstein Generative Adversarial Network [25], which is currently one of the most popular among GANs and is a solution to many problems. However, Wasserstein uses the weight clipping method to prevent the Generative Adversarial Network mode collapse problem. Since this is not a very healthy method, Wasserstein Generative Adversarial Network with Gradient Penalty, which is presented as the development of Wasserstein Generative Adversarial Network presented in another study[26], is used.

In WGAN-GP study unlike the standard Generative Adversarial Networks, the dis-

criminator is named differently. It's called Critic. However, this difference is not only in the name but also in the working logic. This loss function is dependent on an amendment of the GAN scheme in which a discriminator does not classify instances (the "Wasserstein GAN" or the "WGAN"). It gives a number for each instance. This number should not be less than one or larger than 0, so we cannot use 0.5, if an instance is real or fake, as a threshold. Training for discriminators only tries to increase the output for real cases rather than for false instances. Since it cannot really distinguish between true and fake, the discriminator from the WGAN is actually referred to as "critic" rather than "discriminator." This distinction is of theoretical importance, but we can treat it for practical purposes as an awareness that the inputs for the loss functions must not be probabilities.

However, in GAAL-based structures [30],[31], the discriminator plays a leading role as a classifier in the general structure, so simply replacing the discriminator with critic is not correct in terms of anomaly detection and GAAL logic. But it is necessary to take advantage of the advantages of Wasserstein Generative Adversarial Networks. In order to achieve this, as you can see in Figure 4.4, a triple model with both critic, discriminator and generator is set up. If the tasks are to be expressed in this model, the task of the critic is to ensure that the generator produces more realistic data than normal generative adversarial networks, while preventing it from entering the mode collapse problem. In fact, simply to summarize, to bring the pluses of Wassertein GAN with GP [26]. The task of our second network generator is to artificially generate real-like potential anomaly data, thus helping our classifier to best understand the difference between normal and abnormal. Finally, the task of the discriminator is to turn into a classifier in the last case, and to turn into a classifier that detects anomaly by using the boundaries that it learns and draws between fake and real during the training process for normal and abnormal in the testing phase. The fact that the generator, which GAAL-based infrastructures [31] naturally have, produces more realistic and more informative data in every phase of the training, allows the model to be developed by feeding with much more useful data, while providing the development of the model, instead of using useless data for itself. In fact, this is the fact that the selection and training of informative data, which is the basis of active learning, emerged under the GAN structure. The support of the critic in each round and the fact

that the data produced due to the nature of the generator is more informative allows the discriminator to develop continuously. With sufficient training, the discriminator can now be used as an anomaly detection classifier in the testing phase.

CHAPTER 5

EXPERIMENTS

It does not make sense to work on a single dataset in order to prove the anomaly detection and whether the algorithm is at a sufficient level. For this reason, we worked on 8 different datasets for accuracy level tests in our study. Using these 8 different datasets, we tried to prove that our study is not dependent on a single data and can give healthy results in various data. Since it would not be an accurate technique to show the results we obtained by only displaying the scores of our own study, we have created a study that can compare our algorithm with different existing studies. The number of these algorithms tried on different datasets is 8 with our study. To name these algorithms, WGAAL-GP, which is our work, MO-GAAL, SO-GAAL, K nearest neighbor, K-Means, OCSVM, Auto-Encoder and finally Principal Component Analysis. In this way, we enable us to make more accurate and comfortable inferences about what level of results our study can achieve compared to other studies and how successful or unsuccessful the results are. In addition, by testing these 8 different algorithms on 8 different datasets, we can observe the effects of the differences of the datasets on the results. Or, we can examine the behavior of any specific algorithm in datasets with different structures according to the accuracy value. By diversifying the dataset, we avoid the suspicion that our study will yield good results depending on the data. We show results from different datasets and compare these results by sorting between algorithms for each dataset. By collecting the ranking values of the algorithms on different datasets in a table, we enable evaluations only on success.

We also did more specific studies on another dataset. To summarize, these studies progressed as follows. First, we developed the developmental aspects of the GAAL-based anomaly detection study, which our study characterized as fundamental. The

GAAL based anomaly detection study is a structure that exists to detect anomaly with two different basic algorithms. These studies can be expressed as "Single Objective Generative Adversarial Active Learning" and "Multi Objective Generative Adversarial Active Learning". The main difference of these two is that SO-GAAL is trying to produce artificial informative data using a single generator, while MO-GAAL is algorithms that try to produce artificial informative anomaly data using multiple generators. In our work, we train the Generator through Critic using the logic of the Wasserstein Generative Adversarial Network. For this reason, we specifically compared 3 algorithms instead of all algorithms used for anomaly detection in our tests on different metrics on FlightRadar24 data, which we determined as our main dataset. These algorithms were firstly our own work, Wasserstein Generative Adversarial Active Learning with Gradient Penalty (WGAAL-GP), secondly Multi Objective Generative Adversarial Active Learning (MO-GAAL) and finally Single Objective Adversarial Active Learning (SO-GAAL). To summarize the work we have done on these three algorithms, we aimed to increase the reliability and diversity value of the results obtained by activating different metrics instead of working only on accuracy metrics. To specify these metrics in order, we first used the accuracy metric because the most important criterion when calculating the success of anomaly detection is how accurately anomaly data can be detected. We used the precision value as the second metric. Simply put, the precision metric is the metric that expresses how much of what algorithms consider normal is actually normal. The third metric we compared was the recall metric. Summarizing the Recall metric is the metric that expresses how much of the data that should actually be normal is considered normal by the algorithm. Our last metric is the F1 score metric. To summarize this metric simply, we can describe it as the ratio of twice the product of precision and recall values to the sum of precision and recall values. When we compare algorithms on these metrics, we prove on different metrics how our work improves the work we base it on and causes better results.

5.1 Datasets

First of all, a single dataset was not used in this study, so the characteristics of the datasets used are shown in Table 5.1.

- The first dataset used is the Mulcross dataset.This dataset is available for download in a CSV format at OpenML (`https://www.openml.org/d/40897`) Mulcross is a synthetic data set produced with a multivariative normal distribution and containing sufficient number of anomaly data. This dataset consists of 262144 data, and 10 percent of this data is in the class that we will describe as anomaly. In addition, one element of the data consists of 4 different columns.

- Our second dataset is the Ionosphere dataset.This dataset is avaliable for download at UCI machine learning repository (`https://archive.ics.uci.edu/ml/datasets/ionosphere`). This data set has fewer samples, but the anomaly sample rate is much higher. This radar data was collected by a system in Goose Bay, Labrador. In this data, which consists of two different classes, there are two classes, good signals and bad signals. 126 of the Ionosphere dataset, which consists of 351 data, consists of data belonging to the group classified as anomaly. In addition, a data consists of 36 different dimensions.

- Our third dataset is the Arrhythmia dataset.This dataset is avaliable for download at UCI machine learning repository (`https://archive.ics.uci.edu/ml/datasets/arrhythmia`). The aim of this dataset distinguish between the presence and absence of cardiac arrhythmia and classify it in one of the 16 groups. However, the importance of this dataset for our study is that the dimension value is very high compared to other datasets. A data consists of 279 dimensions. In addition, this data set contains 452 samples. To express approximately 15 percent of these samples as numbers, 66 data represents anomaly data.

- Our fourth dataset is the Pima dataset.This dataset is avaliable for download at kaggle (`https://www.kaggle.com/uciml/pima`). This data set can be described as a data set that is very similar to the Ionosphere data set in terms

of distribution. This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. This dataset consists of 768 data and 268 of these data are members of the class to be considered an anomaly. In addition, the data consists of 8 different columns.

- Our fifth data set is our data set called Vertebral.This dataset is avaliable for download at UCI machine learning repository (`http://archive.ics.uci.edu/ml/datasets/vertebral+column`). This dataset can be characterized as a more balanced dataset compared to other data. Data set containing values for six biomechanical features used to classify orthopedic patients into 3 classes (normal, disc hernia, or spondylolisthesis) or 2 classes (normal or abnormal). In this study, we used the second option, normal and abnormal classes. The Vertebral dataset, which consists of 240 data, has an anomaly data rate of 12.5 percentage. This rate shows that 30 data are anomalies. In addition, a data consists of 6 different dimensions.

- Our sixth dataset is our dataset called Shuttle.This dataset is avaliable for download at UCI machine learning repository (`https://archive.ics.uci.edu/ml/datasets/Statlog+(Shuttle)`). When this data set is compared with other data, it becomes the number two data set used in our study in order of sample number. The original Statlog (Shuttle) dataset from UCI machine learning repository is a multi-class classification dataset with dimensionality 9. Here, the training and test data are combined. The smallest five classes, i.e. 2, 3, 5, 6, 7 are combined to form the outliers class, while class 1 forms the inlier class. This data set contains 49097 data and 3511 of these data represent data in the anomaly data class. According to these numbers, about 7 percent of the data can be described as anomaly data. In addition, a data consists of 9 dimensions.

- Our seventh dataset is our dataset called Waveform.This dataset is avaliable for download at UCI machine learning repository (`https://archive.ics.uci.edu/ml/datasets/waveform`) The most important feature of this data set in our tests is that the anomaly data rate is very low. It is a 3 class

classification problem based on 3 waveforms, each of which is sampled at 21 intervals. Each class is a random convex combination of two of the waveforms. The data was generated using David Aha's program from the UCI repository of machine learning databases. This data set contains 3343 data in total. Class 0 member data is defined as anomaly data. The number of these data is about 100. As a percentage, 0.3 percent of the data is considered an anomaly. A data consists of 21 dimensions, which shows that it is a high-dimensional data compared to other data.

- Our eighth dataset is our Stamps dataset.This dataset is avaliable for download at kaggle (`https://www.kaggle.com/rtatman/stamp`). The stamps assure the authenticity of the contents of the documents. The general objective of this dataset is to enable researchers in the field of pattern recognition to analyze, detect, localize and recognize different types of stamps. In this data set, those belonging to the Genuine class are considered as normal data and those belonging to Forged are considered as anomaly data. In total, 31 of 309 data were classified as anomalies. It corresponds to about 10 percent. In addition, a data consists of 9 dimensions.

The summary information of the datasets given above is shown in Table 5.1.

The last data set and the data we can call the main dataset in which detailed studies are carried out is the aircraft tracking data provided by FlightRadar24. After careful literature investigation, we decided to work with the radar track datasets consisting of the ADS-B (ASTERIX CAT21) messages converted FlightRadar24 data. ASTERIX is a EURO-CONTROL air traffic monitoring message format designed for the transmission of information between any monitoring system and automatic monitoring. It defines the structure of the data over a media, from the coding of each piece of data to the arrangement of data in the data block, without any loss of information in all data. Our dataset comprises one day civil flights on 17.10.2019. About one point five million radar tracks are available in our data set. The original data set contains register addresses, latitude, longitude, heading, altitude, speed, squawk code, aircraft type, tail number, departure, arrival flight number and call sign information. However, in this study, we extracted the unique values used to determine the identity of the air

Table 5.1: Dataset Descriptions and Properties

Dataset	Anomaly Percentage	Number of Samples	Number of Anomaly Samples	Dimension
Mulcross	%10	262144	26214	4
Ionosphere	%36	351	126	33
Arrhythmia	%15	452	66	279
Pima	%35	768	268	8
Vertebral	%12,5	240	30	6
Shuttle	%7	49097	3511	9
Waveform	%0.3	3343	100	21
Stamps	%10	309	31	9

trace and worked with the remaining data set. The characteristics of the data that formed the basis of our study were latitude, longitude, heading, altitude and speed parameters. In addition, the data, which is an aircraft type unmanned aerial vehicle, was not used in the training phase. As it can be understood from here, our data is a 5-dimensional data.

By using this data, we wanted to train our model with data containing the kinetic properties of aircraft that can actually be obtained from radars. In this way, we used the data obtained from any other radar to be converted into a format suitable for the model with a little pre-processing.While using this data set, we considered the air tracks in the type of unmanned aircraft as anomaly data. Since it has patterns that are not very similar to civil aircraft, it differs from standard aircraft movements. Therefore, we evaluated unmanned aerial vehicles as anomalies in our dataset. This constitutes approximately 4 % of our total data.

5.2 Results

First, results were obtained on joint datasets on more than one algorithm to decide how accurate anomaly detection was and how successful or unsuccessful it was compared to other algorithms. While obtaining these results, the accuracy metric was used as a criterion. Accuracy, in the simplest terms, is the knowledge of how many percent of the data considered anomaly or normal actually belong to the class that the model says. Mulcross, Ionosphere, Arrhythmia, Pima, Vertebral, Shuttle, Waveform and Stamps datasets, which we mentioned in the previous section, were used. These datasets were tested with MO-GAAL, SO-GAAL, KNN, K-means, OCSVM, AutoEncoder and PCA algorithms, respectively, which are used for anomaly detection. And in addition to these, it was run with the WGAAL-GP algorithm, which is our work, and results were obtained on the basis of accuracy. These algorithms were run on the common dataset without being implemented directly from scratch using the ***Pyod*** library. While testing, the library's default parameter values were used and tested as such. Training and test datasets were kept constant for each algorithm to ensure an objective comparison of the results.We shared the results of all these studies in a table, making it easy to understand and follow. The results on this table are displayed in Table 5.2.

Before presenting the results, we should explain how the different algorithms used in the results are used for anomaly detection and how the results are obtained.

- The algorithm that should be mentioned first is SO-GAAL. SO-GAAL stands for Single Objective Generative Adversarial Active Learning. This algorithm is basically very similar to our work, but it has a structure that uses only Discriminator and Generator, not 3 networks as a structure. In addition, instead of using WGAN-GP as the generative model, he developed the algorithm using the standard GAN model. While making an anomaly decision, anomaly detection is made by using the classifier feature of the sufficiently trained Discriminator, as in our study.

- The second algorithm used is MO-GAAL. This abbreviation means Multi-Objective Adversarial Active Learning. This algorithm is actually a work put

forward to avoid SO-GAAL's problems. The number of Generators in the model has been increased in order to prevent this mode collapse problem, where there is a possible mode collapse problem in SO-GAAL. In this way, although the problem could not be prevented theoretically, it reduced the possibility of its occurrence. In the tests, this model was designed with 3 Generators, which were generally specified as the optimum value in the study. And the results obtained were again obtained by using the classifier feature of Discriminator.

- The third algorithm used KNN for anomaly detection. This means K is the nearest neighbor. In this method, the distances of the newly incoming data with its k closest neighbors are calculated and it is decided which class the incoming data is a member of according to the classes of these neighbors. However, when the average of these distances is taken, if it is above a certain threshold, it can be decided that this point is an anomaly. Because if the distance is far enough, it can mean that the relevant point is not a member of any class. During the tests, we determined the K value as 5 and the results were obtained in this way.

- The fourth algorithm is K-means clustering. Although this method is actually similar to the KNN method, instead of finding its K nearest neighbors, its distances from its clusteroid centers are calculated. After these distances are calculated, the new point actually needs to be considered as a member of the clusteroid closest to it. However, if the distance between the nearest clusteroid and the point is above the predetermined threshold, this point can be considered as not belonging to any cluster. Such a point can be considered as an anomaly. While doing our tests, we tried to detect anomaly with this method. We tried to select the K value by examining the cluster numbers of the data. And after training with data not found in the anomaly class, it was tested with a test data containing normal and anomaly data together.

- The fifth study is OCSVM. This method stands for One Class Support Vector Machine. This method can actually be interpreted as a customized version of the standard SVM method to detect rare events. In the standard SVM technique, the data is tried to be separated into two separate clusters using a hyperplane with maximum margin. Similarly, OCSVM tries to cover the existing data with the smallest margin with a hypersphere. If the new incoming data is

outside of this hypersphere, the data is considered an anomaly. We also used this algorithm in the testing phase in this way and obtained the results from this algorithm.

- The sixth algorithm is Autoencoders, one of the most popular deep learning based methods. Autoencoders are basically used as a method that compresses the data and then tries to reconstruct the data from the jammed state. Calculates the difference between the reconstructed data and the original data. This difference is called reconstruction loss. If this reconstruction loss is greater than the predetermined threshold value, this data can be considered an anomaly. We also detected anomaly classes in this way while testing in our study.

- PCA was used as the final algorithm. This method stands for principal component analysis. Although this method is logically similar to Autoencoders, it provides a more primitive approach in terms of working logic. The data is transferred to the plane in a way that can be expressed with smaller dimensions. If it tries to express the new data in the same way, and if the data re-expressed on the plane reveals much larger loss values compared to other data, this data can be considered as an anomaly. The size of this loss value is determined by comparing it with the previously determined threshold value. In this study, we performed anomaly detection with PCA in this way.

Table 5.2: Experimental Results of Anomaly Detection Methods on Different Datasets

Datasets	WGAAL-GP	MO-GAAL	SO-GAAL	KNN	K-Means	OCSVM	AutoEncoder	PCA
Mulcross	**0.913**	0.904	0.889	0.839	0.831	0.784	0.911	0.707
Ionosphere	0.867	0.829	0.729	0.919	**0.924**	0.747	0.875	0.739
Arrhythmia	**0.744**	0.711	0.689	0.708	0.703	0.661	0.699	0.462
Pima	0.722	**0.733**	0.644	0.706	0.655	0.544	0.724	0.514
Vertebral	0.819	0.844	0.712	0.811	0.821	0.512	**0.872**	0.578
Shuttle	0.909	0.902	0.891	**0.920**	0.917	0.651	0.887	0.633
Stamps	**0.917**	0.899	0.644	0.901	0.868	0.702	0.914	0.681
Waveform	**0.841**	0.821	0.826	0.766	0.731	0.585	0.824	0.591

The first dataset tried was the Mulcross dataset. The results obtained on this dataset were higher in percentage terms when compared to the others. When the algorithms

are compared, it is noteworthy that the most successful algorithms are deep learning based algorithms. But it seems that WGAAL-GP achieved the highest result. In second place is the Ionosphere data. Here, the most successful algorithm is the K-Means algorithm, followed by the KNN algorithm. This shows that distance-based clustering algorithms are more useful on this data, but our study also gets a good result. If the next dataset is Arrhythmia data, WGAAL-GP is again the algorithm that gives the highest result. The fact that the MO-GAAL algorithm is in the second place reveals how effective the algorithms that pass the training process by producing artificial anomaly data are for this data. When the results on the Pima dataset, which is another dataset, are examined, it is seen that the most successful study is the MO-GAAL algorithm, but it is right behind it with a very small difference in our study. Considering the results of experiments on another dataset, the Vertebral dataset, it is seen that reconstruction-based algorithms are more successful. In this context, algorithms that learn the structure of the data better come to the fore. Although the Autoencoder gave the best result, it still achieves a result that is in the top 3 in WGAAL-GP. When the results on the Shuttle dataset are examined, it is seen that distance-based clustering algorithms give good results, but even for this dataset, WGAAL-GP is ahead of other algorithms, just behind K-means, and gives the best results among the rest. When the results for Stamps data, another frequently used data set, are examined, it is seen that our study is the algorithm with the highest accuracy value. It was the study that detected the highest percentage of anomaly among 8 algorithms for this data. For the last dataset, Waveform data, WGAAL-GP is the algorithm with the highest accuracy value, and when evaluated in general, it is roughly in the top 3 for each data set and has the highest accuracy values on average when compared to the others. Again, a more complete summary of all these results can be seen in Table 5.2.

When we evaluated the results on the basis of accuracy, specific to datasets and algorithms, the results in Table 5.2 were obtained. By sorting these results among the algorithms separately for each dataset, creating a success ranking among the algorithms increases the interpretability of the results. The information on which this ranking is collected can be seen in Table 5.3.

When the table is examined, WGAAL-GP has the highest accuracy result in 4 out of 8 datasets. It showed a very successful result as the 4th worst among other datasets. On

Table 5.3: Ranking Comparison between Anomaly Detection Algorithms on Different Datasets

Metric	Rank							
Datasets	WGAAL-GP	MO-GAAL	SO-GAAL	KNN	K-Means	OCSVM	AutoEncoder	PCA
Mulcross	**1**	3	4	5	6	7	2	8
Ionosphere	4	5	8	2	**1**	6	3	7
Arrhythmia	**1**	2	6	3	4	7	5	8
Pima	3	**1**	6	4	5	7	2	8
Vertebral	4	2	6	5	3	8	**1**	7
Shuttle	3	4	5	**1**	2	7	6	8
Stamps	**1**	4	8	3	5	6	2	7
Waveform	**1**	4	2	5	6	8	3	7
Average Ranks	**2.25**	3.12	6.37	3.50	4.12	7.00	3.0	7.50

average, it was the algorithm with the lowest average ranking among other algorithms with 2.25. MO-GAAL study, which is likely to be the biggest competitor, with 3.12 average ranking and Autoencoder study with 3.0 average ranking were also the closest algorithms in the results. Among other algorithms, although KNN is a distance-based algorithm, it achieved a very close result to deep learning based algorithms with an average ranking of 3.5. However, as a result, WGAAL-GP was the algorithm that gave the most successful results among 8 different algorithms with 8 different datasets on the basis of accuracy.

Examining this table, it seems that the algorithm with the highest average ranking is our study WGAAL-GP. There are different factors that put this work ahead of other algorithms. Since these factors are examined, the first factor is to learn the properties of the data with a deep learning-based technique and work with a structure built through reconstruction. In this way, the model learns more information about the data compared to distance-based or linear-based algorithms. Among the deep learning algorithms, the most important feature that makes it stand out is that it is a generative algorithm, so it is not only satisfied with the data provided by the dataset during the training process, but also provides more opportunity for the development of the model by producing artificial data. Especially when GAAL based deep learning algorithms, namely WGAAL-GP, MO-GAAL and SO-GAAL are examined, the biggest advantage against Autoencoder is these generative structures. Because, especially when a

problem such as anomaly detection is examined, the number of anomaly data is very limited in normal data, which is one of the biggest obstacles to training the model well during the training process. However, AGPO based methods avoid this limited data and training problem by generating potential anomaly data. In addition, GAAL based methods make the quality and production of these potential anomaly data produced by blending this with active learning more intelligently than randomness. Finally, the biggest reason why our study got higher results compared to other GAAL-based structures is that it uses Wasserstein GAN instead of using standard GAN. In this way, it already has a more advanced Generator compared to the standard GAN Generator. This Generator is therefore able to generate higher quality and informative potential anomaly data. And the model is making more significant strides by being trained with more informative data. In addition, since it gets rid of the mode collapse problem, which is the most important of the possible GAN problems, with WGAN, for example, it does not allow the SO-GAAL algorithm to get a very low result compared to the others in Stamps data.

While evaluating the test results, comparisons were made on Precision, Recall, Accuracy and F1 score metrics. How these metrics are calculated and with what values they are formed are available in the explanations below. These metrics ensure that the data is not evaluated only on accuracy values, and more realistic and accurate comparisons are made by obtaining results on different metrics.

$$Precision = \frac{TP}{TP + FP} \qquad F1score = \frac{2 * Precision * Recall}{Precision + Recall}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \qquad Recall = \frac{TP}{TP + FN}$$

True Positives (TP) : Number of correct normal labeled data.

True Negatives (TN) : Number of correct labeled abnormal labeled data.

False Positives (FP) – Number of incorrect normal labeled data.

False Negatives (FN): Number of incorrect abnormal labeled data.

Table 5.4: Precision Results of Anomaly Detection Methods on Different Datasets

Datasets	WGAAL-GP	MO-GAAL	SO-GAAL	KNN	K-Means	OCSVM	AutoEncoder	PCA
Mulcross	**0.852**	0.826	0.819	0.779	0.791	0.724	0.851	0.650
Ionosphere	0.811	0.799	0.621	0.888	**0.901**	0.641	0.789	0.599
Arrhythmia	**0.699**	0.611	0.582	0.605	0.601	0.559	0.597	0.392
Pima	**0.687**	0.685	0.574	0.601	0.535	0.414	0.629	0.441
Vertebral	0.731	0.724	0.610	0.734	0.692	0.442	**0.751**	0.487
Shuttle	0.904	0.883	0.878	**0.913**	0.865	0.581	0.792	0.573
Stamps	**0.901**	0.829	0.547	0.809	0.818	0.632	0.857	0.593
Waveform	**0.802**	0.761	0.791	0.752	0.701	0.595	0.784	0.601
Average Ranks	1.50	3.12	5.25	3.50	4.50	7.12	3.57	7.50

Table 5.5: Recall Results of Anomaly Detection Methods on Different Datasets

Datasets	WGAAL-GP	MO-GAAL	SO-GAAL	KNN	K-Means	OCSVM	AutoEncoder	PCA
Mulcross	**0.814**	0.802	0.791	0.735	0.732	0.681	0.808	0.601
Ionosphere	0.759	0.731	0.626	**0.824**	0.814	0.657	0.769	0.641
Arrhythmia	**0.746**	0.691	0.673	0.689	0.623	0.591	0.638	0.492
Pima	**0.713**	0.709	0.577	0.616	0.575	0.449	0.694	0.483
Vertebral	0.721	0.729	0.622	0.727	0.751	0.502	**0.793**	0.528
Shuttle	0.893	0.881	0.874	**0.902**	0.899	0.631	0.871	0.644
Stamps	0.899	0.879	0.624	0.866	0.848	0.691	**0.903**	0.687
Waveform	**0.803**	0.791	0.794	0.742	0.703	0.572	0.785	0.602
Average Ranks	2.25	3.12	5,37	3.75	4.25	7.25	3.12	7.25

In addition, it is seen that our study has made a significant difference to its competitors in precision values, while it has more similar results in recall values. When this situation is examined, first of all, it should be understood what exactly the precision and recall metrics mean. Precision data expresses how many percent of what the model classifies as normal is actually normal. The Recall metric is the metric that indicates what percentage of truly normal data is classified as normal by the model. Our work, on the other hand, tries to create the narrowest lines where all of the normal data is covered by producing more realistic data. In this way, more anomalies allow data to be left out. However, while this actually increases the precision of the model, it does not improve recall when compared to more roughly drawn boundaries. Therefore, while it makes a significant difference to other algorithms in terms of precision, it is normal to get more similar results when examined in the recall metric.

Table 5.6: F1 Score Results of Anomaly Detection Methods on Different Datasets

Datasets	WGAAL-GP	MO-GAAL	SO-GAAL	KNN	K-Means	OCSVM	AutoEncoder	PCA
Mulcross	**0.832**	0.813	0.804	0.756	0.760	0.701	0.828	0.624
Ionosphere	0.784	0.763	0.623	0.854	**0.855**	0.648	0.778	0.619
Arrhythmia	**0.746**	0.648	0.624	0.644	0.611	0.574	0.616	0.436
Pima	**0.699**	0.696	0.575	0.608	0.554	0.445	0.659	0.461
Vertebral	0.725	0.726	0.615	0.730	0.720	0.470	**0.771**	0.506
Shuttle	0.898	0.881	0.875	**0.907**	0.881	0.631	0.829	0.606
Stamps	**0.900**	0.853	0.582	0.836	0.832	0.660	0.879	0.636
Waveform	**0.802**	0.775	0.799	0.733	0.707	0.552	0.788	0.604
Average Ranks	1.75	3.12	5,12	3.37	4.75	7.12	3.25	7.50

It is seen that our study was generally successful, but it has lower ranking values in some datasets compared to others. To understand the reasons for this, it is necessary to examine the datasets. Ionosphere and Pima are examples of these datasets. When we look at the common features of these data, it is seen that the data with the highest anomaly data rate among the benchmark datasets. This rate constitutes approximately one third of the data. Normally, the anomaly rates in the data for anomaly detection are at lower percentages. In cases where the anomaly rate is this high, other algorithms can also achieve high results. Because finding anomalies on the data becomes an easier problem. However, when we look at the datasets with lower anomaly rate, we can see that WGAAL-GP is much more successful, and it is clearly stated in the tables that it has the most successful results in most of them. This happens because we produce much more realistic potential anomaly data, and our classifier can draw sharper and clearer boundaries with clearer information about the data.

We conducted tests on FlightRadar24 data more detailed version with this metrics. In this study, we show whether an improvement has been made on existing GAAL structures by comparing with GAAL based anomaly detection algorithms instead of all anomaly detection algorithms. Because we basically offer solutions to the weaknesses of these existing studies or to the problems in the areas that can be improved. To summarize, these tests compare our work (WGAAL-GP) and a MO-GAAL model designed with three Generators and SO-GAAL algorithms.

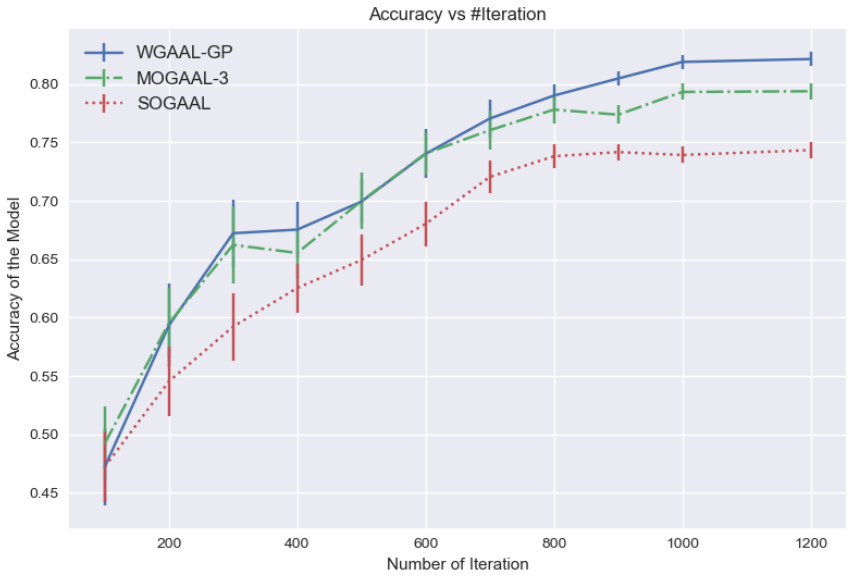

Figure 5.1: Accuracy scores for GAAL algorithms according to number of training samples

Firstly, the results obtained for the accuracy metric to be compared with these algorithms are displayed in Figure 5.1. The graph here expresses the number of traning samples on the x-axis and the accuracy score on the y-axis. While WGAAL-GP and MO-GAAL accuracy results were similar to each other in the first phase of the training, that is, up to roughly 600,000 data, it seems that WGAAL-GP got more successful results in the second phase of the training process. It is seen that the SO-GAAL algorithm gives results behind compared to the other two algorithms. The main reason for this situation is that the potential anomaly data produced in the first process of training is more crude and very unrealistic potential data, so Discriminator cannot create its boundaries with sufficient precision when classifying. However, the potential anomaly data produced in the later phases of training begins to become closer to the real data and the Discriminator begins to draw its boundaries closer to the real data. In this process, WGAAL-GP is able to generate more informative potential anomaly data as it improves its Generator thanks to Critic. In this way, Discriminator draws its boundaries sharper and more accurately. As a result, as the WGAAL-GP number of samples increased, the accuracy difference between MO-GAAL increased.As a result, WGAAL-GP was the most successful study in the tests performed on the

relevant dataset on the basis of accuracy.

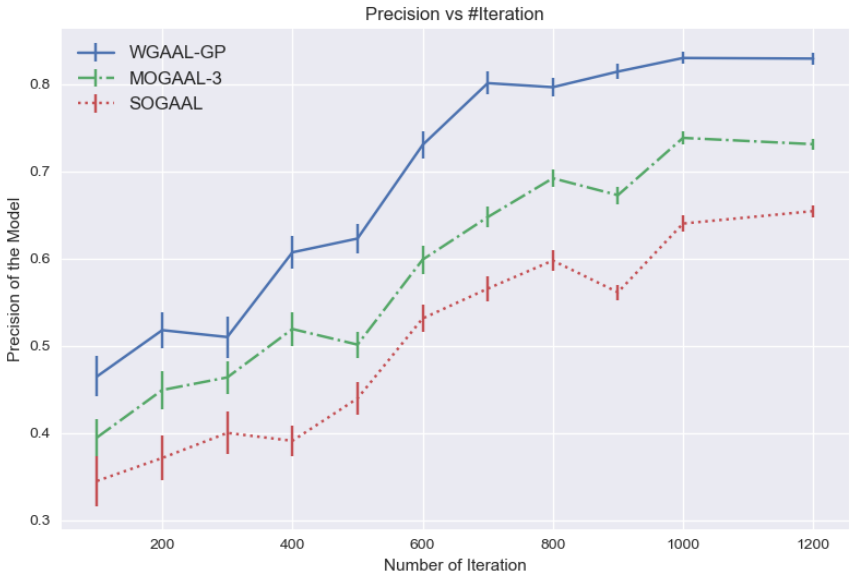

Figure 5.2: Precision scores for GAAL algorithms according to number of training samples

To make an evaluation within the scope of Precision, as can be clearly seen in Figure 5.2, WGAAL-GP achieved much higher and more successful results from the beginning to the end of the training process compared to the other two algorithms. To understand getting these results, one must first understand exactly what the Precision metric represents. Precision expresses what percentage of the data that the model qualifies as normal is actually normal. In fact, this situation can be thought of as such in the case of GAN. The boundaries drawn by the Discriminator are actually an area that tries to include all normal data with the narrowest possible limits. While drawing these boundaries, the most important factor is actually the potential anomaly data produced by the Generator, because the closer these data are to the truth, the closer the boundary tried to be drawn will be to the truth. In fact, the number one novelty of the WGAAL-GP study is that by training the Generator with Critic, it produces more realistic data than standard GAN models. Therefore, when Figure 5.2 is examined, it can be interpreted as a graphical reflection of the fact that the boundaries of the Discriminator are sharper and closer to reality. As a result, it is shown that the data classified as normal by the WGAAL-GP study are classified with a higher percent-

age than other studies, and the boundaries drawn for the normal classification by the Discriminator are more accurate.

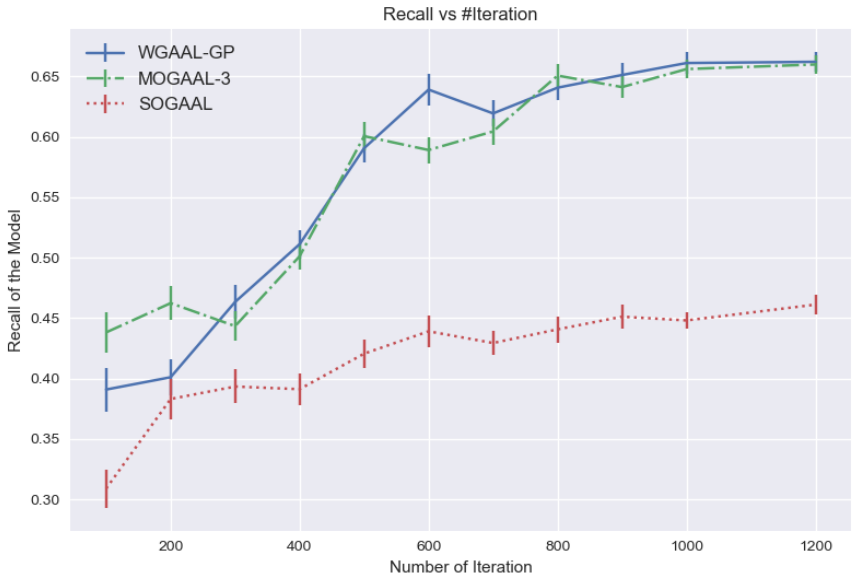

Figure 5.3: Recall scores for GAAL algorithms according to number of training samples

The comparison between WGAAL-GP, MO-GAAL and SO-GAAL in recall values, which is another different metric, is shown in figure 5.3. First of all, it should be understood more clearly what the Recall metric actually means. In fact, this can be evaluated as follows, how much of the data that is actually normal, the model has qualified as really normal data. Again, when this situation is examined in the perspective of GAN and anomaly detection, it can be expressed as follows. The area outside the boundaries drawn by the discriminator is normally considered as anomaly data. However, normal data in this region may be evaluated as abnormal due to the fact that the borders are not drawn correctly. This situation actually decreases the recall score and this situation is shown in figure 5.3. When the figure is examined, it is seen that the SO-GAAL algorithm achieves much lower recall values compared to the other two algorithms. This may be due to the fact that the structure using a single standard Generator does not provide enough diversity or is open to the mode collapse problem. When the other two algorithms are examined, it is seen that WGAAL-GP has a more accelerated graph when it is evaluated as a learning curve in the training

process. However, when the process at the very beginning of the training is excluded, it is seen that although MO-GAAL is ahead of WGAAL-GP at two different points, in other cases each WGAAL-GP achieves higher scores. Although the results are close to each other when the data is fully trained, it is seen that WGAAL-GP achieves higher scores. Therefore, in general, it can be said that the recall metric of WGAAL-GP is the most successful algorithm on the relevant data.

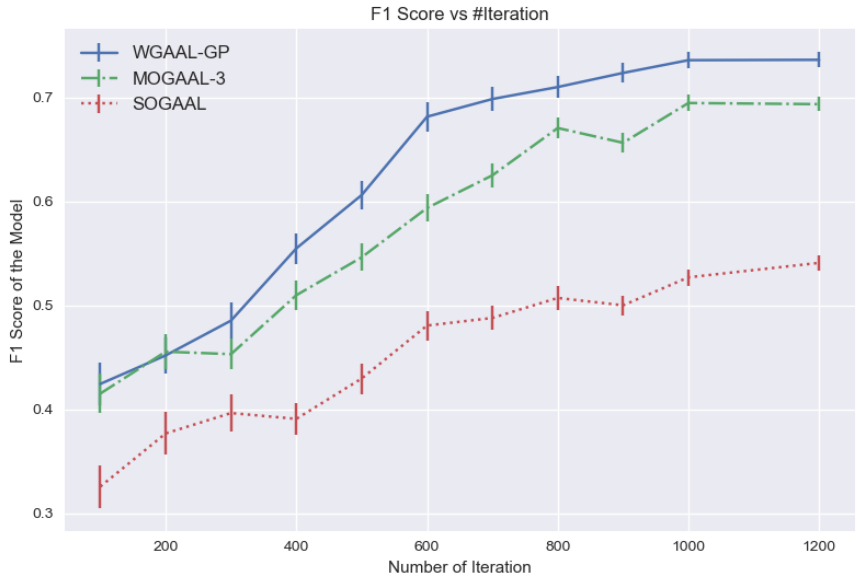

Figure 5.4: F1 scores for GAAL algorithms according to number of training samples

Our another metric is F1 score. F1 score and number of samples graph are displayed in figure 5.4 with the results obtained for GAAL based algorithms. Again, the first striking situation on the graph is that the SO-GAAL algorithm lags far behind the other two algorithms. When the other two algorithms are examined, it is seen that only the MO-GAAL algorithm is ahead of WGAAL-GP in a part of the very early stages of training. Apart from that, the WGAAL-GP has a higher F1 score than other studies in absolute terms. When Figure 5.2 and Figure 5.3 are examined, it is seen that they are ahead in precision and recall metrics. It is a metric obtained by mathematically proportioning the precision and recall metrics while calculating the F1 score. Therefore, the most successful result of WGAAL-GP as an F1 score is a natural result when interpreted together with other metrics.

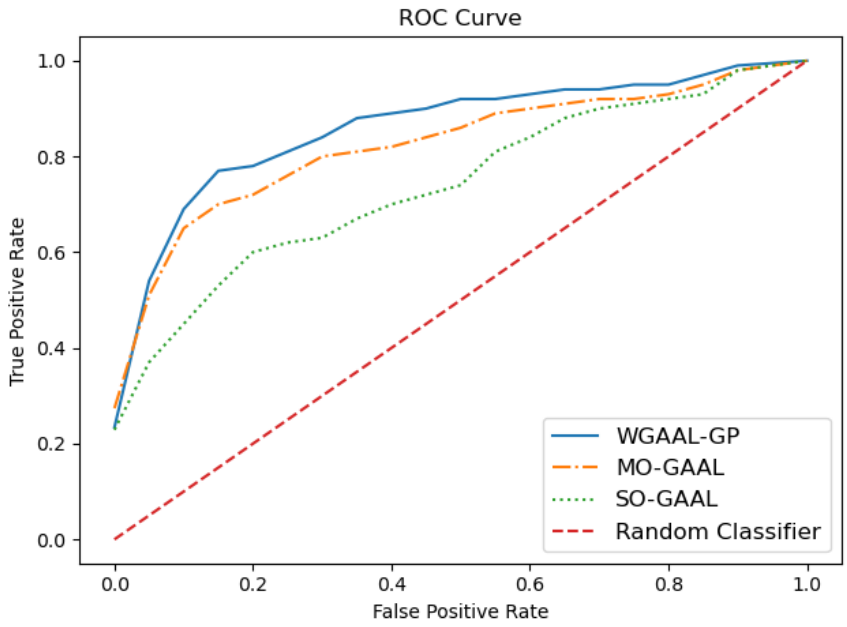

Figure 5.5: ROC Curves for GAAL algorithms

Our final metric is the ROC curve. ROC stands for receiver operating characteristic. A data showing the correct decision making capabilities of this curve classifier and how it labels data with these capabilities. It is defined over false positive rate and true positive rate at different threshold values. Before interpreting the ROC curve, let us state how the superiority among algorithms is demonstrated in this metric. While interpreting this metric, there is a random classifier line as you can see in Figure 5.5. The graphs above this line indicate that the algorithm makes smarter decisions instead of randomness. The superiority levels of these algorithms are explained by looking at the areas under the curve. The larger the area under the curve, the more successful the classifier means. When Figure 5.5 is examined, it is seen that the curve of WGAAL-GP has a higher AUC (Area Under the Curve) score compared to the other two GAAL-based studies. This shows that our study is more successful in this metric compared to other studies.

To summarize, it is seen that WGAAL-GP is the most successful algorithm in terms of average ranking as a result of the tests performed on different datasets by comparing them with different algorithms only in terms of accuracy. In addition, precision, recall and F1 score metrics were compared with GAAL studies, which are the ba-

sis of our work. As a result of these comparisons, it was seen that the WGAAL-GP study was ahead of other GAAL-based anomaly detection studies. The main reason for this is the difference between the distributions by using KL distance in the standard GAN structure, but Wasserstein uses EM distance instead of GAN KL distance. In this way, it can give much better feedback for the Generator. The quality of the potential anomaly data produced by the standard GAN is poor compared to the potential anomaly data produced by the WGAAL-GP. This makes it even easier for the Discriminator side of the model to distinguish between real and fake data because it is exposed to more informative data. In addition, since other algorithms use standard GAN, it is vulnerable to mode collapse problem. They may have had such problems from different datasets and yielded lower results. However, since WGAAL-GP uses Wasserstein GAN and one of the main advantages of this study is that it avoids the mode collapse problem, WGAAL-GP can achieve higher scores.

Another important thing that our study promises is that it avoids the mode collapse problem. In order to identify the mode collapse problem, the way referenced in the literature is to examine the outputs and to examine whether the diversity is really lost. Another way is to examine the loss graphs of the models. Since there are 3 different models in the WGAAL-GP structure, the loss records of all 3 models are kept and graphed from the iteration measure. In fact, the training process, which ranges from 0 to 1400 iterations, ranges from 0 to 7000 iterations for the Critic network. The reason for this is that Wasserstein is run more in the GAN structure than the Critic Generator, and this number of studies is determined parametrically. In our study, we determined this number as 5, so Critic 7000 iteration actually worked for 1400 iterations. The Generator loss chart can be seen in figure 5.5, the Discriminator loss chart in figure 5.6 and finally the Critic loss chart in figure 5.7.

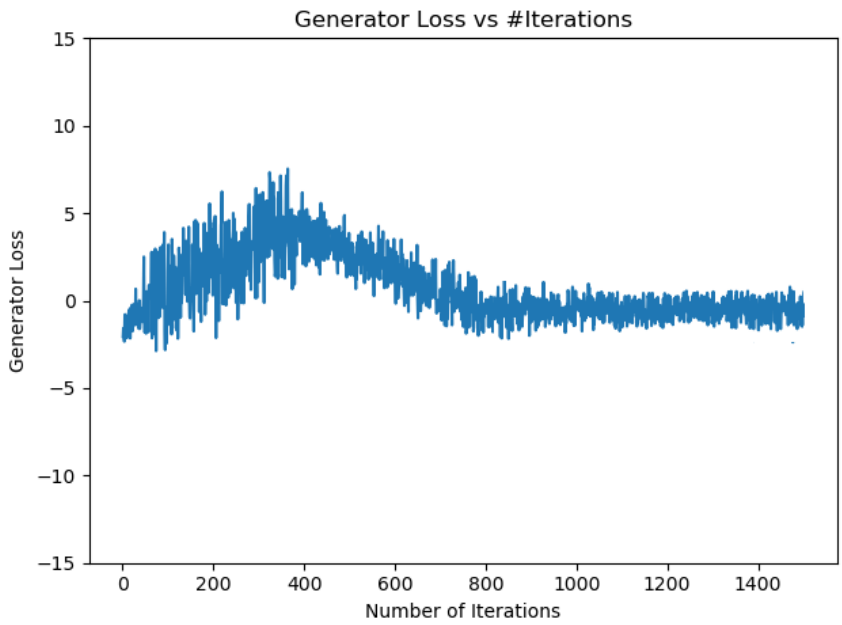

Figure 5.6: Generator loss for WGAAL-GP algorithm

When we examine these loss charts, there is a lot of fluctuation in the first phase of Generator training. However, when we take the center of the fluctuations as a basis, it is seen that the loss values increase first. However, when the process progresses a little further, it is seen that both the difference value of the fluctuations decreased noticeably and the loss center began to decrease. At the end of the training, it is seen that the loss value converges to zero and the fluctuations are much smaller. As a result, a graph suitable for a healthy Generator loss pattern was obtained. If there were findings such as mode collapse etc, there should be places where the level and intensity of fluctuations increased visibly.

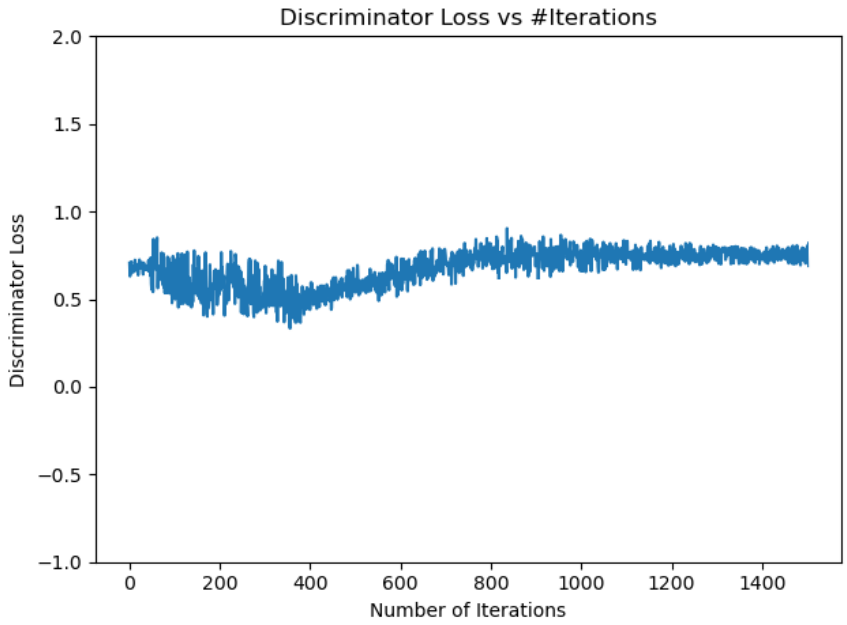

Figure 5.7: Discriminator loss for WGAAL-GP algorithm

When we examine the loss graph of the Discriminator, we can actually interpret the graph of the Generator as symmetrical or very similar. While the intensity of the fluctuations was very high at the beginning of the training, the intensity of the fluctuations towards the end of the training became much more reasonable. At the same time, although the loss value started to decrease at first, it increased a little later and changed between 0.5 and 1. Again, a result with suspicion of mode collapse does not appear and behaves like the expected Discriminator loss chart.

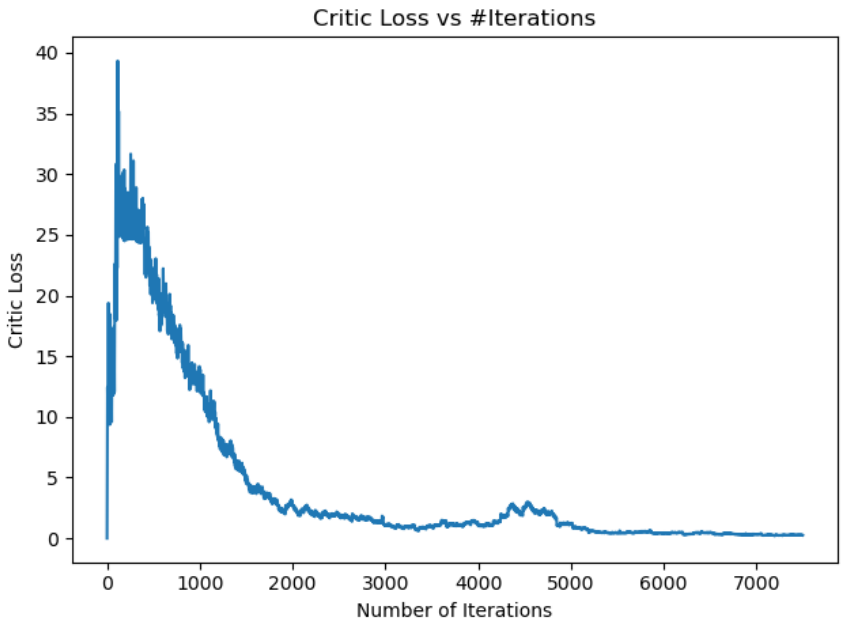

Figure 5.8: Critic loss for WGAAL-GP algorithm

Finally, when the Critic loss chart is examined, it shows a very aggressive rise at the beginning. But then, with the development of the Generator, it started to balance and started to decrease gradually after the peak point. Again, while the fluctuations were very severe at the beginning, as the number of iterations increased, it reached an acceptable level of fluctuation intensities. Then the loss curve started to converge to zero. In the process of reaching zero, while the fluctuation intensities were higher at first, it reached equilibrium around zero with very little fluctuation in 5000 iterations and afterwards.

As a result, WGAAL-GP was the algorithm with the highest average ranking on the basis of accuracy with different algorithms on different datasets. It also achieved higher scores than SO-GAAL and MO-GAAL in precision, recall and F1 score metrics on FlightRadar24 data. This means that it has developed these algorithms and can detect anomalies with a higher rate. In addition, the loss graphs of the models are shown due to the mode collapse problem, which is another problem it solves. Nothing out of the ordinary was observed in these loss charts and no mode collapse indicator was found. In other words, this study showed that the two novelties it promised, detecting anomalies with a higher success rate and preventing the mode collapse prob-

lem, on the analyzed data. By using Wasserstein GAN, the potential anomaly data produced in GAAL-based studies was of higher quality, thus achieving higher success rates. At the same time, due to the nature of the Wasserstein GAN, it also prevented the mode collapse problem. Our study also showed these situations with its results.

CHAPTER 6

CONCLUSIONS

In this thesis, an improvement and development is proposed in which GAAL-based studies used for anomaly detection can be achieved with higher success and will be protected from the mode collapse problem. One of the biggest obstacles when detecting anomaly is the very limited number of anomaly data. In order to prevent this problem, potential anomaly data can be produced thanks to generative models. This raises the number of anomaly data to an acceptable level and facilitates models that detect anomaly. However, it is not enough to increase the anomaly data numerically, but it is also necessary to increase its informativeness. In this study, the main focus is to make the model more successful by increasing the informative level of the potential anomaly data produced. This thesis study proposes as a novel to produce more informative data than existing systems by producing potential anomaly data using Wasserstein GAN with gradient penalty instead of using standard GAN directly. By using WGAAL-GP Wasserstein GAN, it not only produces more informative potential anomaly data, but also avoids the mode collapse problem. In this way, more stable results can be obtained in multidimensional data, and since it does not experience mode collapse in datasets with a high number of clusters, it can improve itself with more diverse potential anomaly data and achieve higher results. This study is compared with different algorithms that have been proven on FlightRadar24 data and some real world datasets, and its results are presented.

Recently, time-critical situations have become too much when evaluating data. Therefore, the time information took a very important place in the data and it became more important to evaluate it together with the past data. When anomaly detection is required on time series data, the most important situation is the correct use of historical

data. It is considered as a target for our future studies to redesign the model structures in an integrated way with LSTM so that our study can handle time series data better. In this way, our generative model can generate much more reliable potential anomaly data for time series data by adding historical data to the evaluation. In addition, in order to shorten the training process, it can be applied to train with a smaller subset by sorting the potential anomaly data produced in each epoch according to the informative level. In this way, the training time of the model for future works can be reduced.

REFERENCES

[1] Dave, D., & Varma, T. (2014). A Review of various statistical methods for Outlier Detection. International Journal of Computer Science & Engineering Technology (IJCSET), 5(2), 137-140.

[2] Kaushik, S., Choudhury, A., Dasgupta, N., Natarajan, S., Pickett, L., & Dutt, V. (2019). Evaluating Auto-encoder and Principal Component Analysis for Feature Engineering in Electronic Health Records. In ITISE 2019. Proceedings of papers. Vol 2.

[3] Pimentel, T., Monteiro, M., Viana, J., Veloso, A., & Ziviani, N. (2018). A generalized active learning approach for unsupervised anomaly detection. stat, 1050, 23.

[4] Vapnik, V. (2013). The nature of statistical learning theory. Springer science & business media.

[5] Hayton, P., Schölkopf, B., Tarassenko, L., & Anuzis, P. (2000, December). Support vector novelty detection applied to jet engine vibration spectra. In NIPS (pp. 946-952).

[6] Ramaswamy, S., Rastogi, R., & Shim, K. (2000, May). Efficient algorithms for mining outliers from large data sets. In Proceedings of the 2000 ACM SIGMOD international conference on Management of data (pp. 427-438).

[7] Piciarelli, C., & Foresti, G. L. (2007, September). Anomalous trajectory detection using support vector machines. In 2007 IEEE Conference on Advanced Video and Signal Based Surveillance (pp. 153-158). IEEE.

[8] Kriegel, H. P., Schubert, M., & Zimek, A. (2008, August). Angle-based outlier detection in high-dimensional data. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 444-452).

[9] Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000, May). LOF: identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD international conference on Management of data (pp. 93-104).

[10] Taha, A., & Hadi, A. S. (2019). Anomaly detection methods for categorical data: A review. ACM Computing Surveys (CSUR), 52(2), 1-35.

[11] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3), 1-58.

[12] Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. Journal of the royal statistical society. series c (applied statistics), 28(1), 100-108.

[13] Das, S., Matthews, B. L., Srivastava, A. N., & Oza, N. C. (2010, July). Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 47-56).

[14] Jolliffe, I. T. (2002). Springer series in statistics. Principal component analysis, 29.

[15] Günter, S., Schraudolph, N., & Vishwanathan, S. (2007). Fast iterative kernel principal component analysis.

[16] Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. AIChE journal, 37(2), 233-243.

[17] Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

[18] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.

[19] Li, L. (2013). Anomaly detection in airline routine operations using flight data recorder data (Doctoral dissertation, Massachusetts Institute of Technology).

[20] Callegari, C., Donatini, L., Giordano, S., & Pagano, M. (2018). Improving stability of PCA-based network anomaly detection by means of kernel-PCA. International Journal of Computational Science and Engineering, 16(1), 9-16.

[21] Bergman, L., Cohen, N., & Hoshen, Y. (2020). Deep nearest neighbor anomaly detection. arXiv preprint arXiv:2002.10445.

[22] Tuluptceva, N., Bakker, B., Fedulova, I., Schulz, H., & Dylov, D. V. (2020). Anomaly detection with deep perceptual autoencoders. arXiv preprint arXiv:2006.13265.

[23] Schlegl, T., Seeböck, P., Waldstein, S. M., Schmidt-Erfurth, U., & Langs, G. (2017, June). Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In International conference on information processing in medical imaging (pp. 146-157). Springer, Cham.

[24] Li, D., Chen, D., Goh, J., & Ng, S. K. (2018). Anomaly detection with generative adversarial networks for multivariate time series. arXiv preprint arXiv:1809.04758.

[25] Arjovsky, M., Chintala, S., & Bottou, L. (2017, July). Wasserstein generative adversarial networks. In International conference on machine learning (pp. 214-223). PMLR.

[26] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). Improved training of wasserstein gans. arXiv preprint arXiv:1704.00028.

[27] Zenati, H., Foo, C. S., Lecouat, B., Manek, G., & Chandrasekhar, V. R. (2018). Efficient gan-based anomaly detection. arXiv preprint arXiv:1802.06222.

[28] Donahue, J., Krähenbühl, P., & Darrell, T. (2016). Adversarial feature learning. arXiv preprint arXiv:1605.09782.

[29] Akcay, S., Atapour-Abarghouei, A., & Breckon, T. P. (2018, December). Ganomaly: Semi-supervised anomaly detection via adversarial training. In Asian conference on computer vision (pp. 622-637). Springer, Cham..

[30] Liu, Y., Li, Z., Zhou, C., Jiang, Y., Sun, J., Wang, M., & He, X. (2019). Generative adversarial active learning for unsupervised outlier detection. IEEE Transactions on Knowledge and Data Engineering, 32(8), 1517-1528.

[31] Zhu, J. J., & Bento, J. (2017). Generative adversarial active learning. arXiv preprint arXiv:1702.07956.

[32] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. Advances in neural information processing systems, 29, 2234-2242.

[33] Haloui, I., Gupta, J. S., & Feuillard, V. (2018). Anomaly detection with Wasserstein GAN. arXiv preprint arXiv:1812.02463.

[34] Zhang, S., Li, X., Zong, M., Zhu, X., & Wang, R. (2017). Efficient kNN classification with different numbers of nearest neighbors. IEEE transactions on neural networks and learning systems, 29(5), 1774-1785.

[35] Jonahtan Hui. (June 14 2018). Gan Wasserstein Gan, WGAN GP URL: https://jonathan-hui.medium.com/gan-wasserstein-gan-wgan-gp-6a1a2aa1b490

[36] Diederik P. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[37] Aksel Wilhelm Wold Eide, A. W. W. (2018). Applying generative adversarial networks for anomaly detection in hyperspectral remote sensing imagery (Master's thesis, NTNU).

[38] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3), 1-58.

[39] Chandola, V., Banerjee, A., & Kumar, V. (2010). Anomaly detection for discrete sequences: A survey. IEEE transactions on knowledge and data engineering, 24(5), 823-839.

[40] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[41] Zhang, J., Zheng, Y., Qi, D., Li, R., & Yi, X. (2016, October). DNN-based prediction model for spatio-temporal data. In Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems (pp. 1-4).

[42] Zhang, J., Zheng, Y., & Qi, D. (2017, February). Deep spatio-temporal residual networks for citywide crowd flows prediction. In Thirty-first AAAI conference on artificial intelligence.

[43] Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., ... & Li, Z. (2018, April). Deep multi-view spatial-temporal network for taxi demand prediction. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1).

[44] Yao, H., Tang, X., Wei, H., Zheng, G., & Li, Z. (2019, July). Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 5668-5675).

[45] Yao, H., Tang, X., Wei, H., Zheng, G., Yu, Y., & Li, Z. (2018). Modeling spatial-temporal dynamics for traffic prediction. arXiv preprint arXiv:1803.01254.

[46] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

[47] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems, 30.

[48] Durugkar, I., Gemp, I., & Mahadevan, S. (2016). Generative multi-adversarial networks. arXiv preprint arXiv:1611.01673.

[49] Zhang, Z., Song, Y., & Qi, H. (2018, March). Decoupled learning for conditional adversarial networks. In 2018 IEEE Winter Conference on Applications of Computer Vision (WACV) (pp. 700-708). IEEE.

[50] Khrulkov, V., & Oseledets, I. (2018, July). Geometry score: A method for comparing generative adversarial networks. In International Conference on Machine Learning (pp. 2621-2629). PMLR.

[51] Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196.

[52] Lucic, M., Kurach, K., Michalski, M., Gelly, S., & Bousquet, O. (2017). Are gans created equal? a large-scale study. arXiv preprint arXiv:1711.10337.

[53] Snell, J., Ridgeway, K., Liao, R., Roads, B. D., Mozer, M. C., & Zemel, R. S. (2017, September). Learning to generate images with perceptual similarity metrics. In 2017 IEEE International Conference on Image Processing (ICIP) (pp. 4277-4281). IEEE.

[54] White, T. (2016). Sampling generative networks: Notes on a few effective techniques. CoRR, abs/1609.04468, 7.

[55] Fernandes, G., Rodrigues, J. J., Carvalho, L. F., Al-Muhtadi, J. F., & Proença, M. L. (2019). A comprehensive survey on network anomaly detection. Telecommunication Systems, 70(3), 447-489.

[56] Wang, Z., & Oates, T. (2015, April). Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. In Workshops at the twenty-ninth AAAI conference on artificial intelligence.

[57] Chandola, V., Banerjee, A., & Kumar, V. (2010). Anomaly detection for discrete sequences: A survey. IEEE transactions on knowledge and data engineering, 24(5), 823-839.

[58] Kamnitsas, K., Baumgartner, C., Ledig, C., Newcombe, V., Simpson, J., Kane, A., ... & Glocker, B. (2017, June). Unsupervised domain adaptation in brain lesion segmentation with adversarial networks. In International conference on information processing in medical imaging (pp. 597-609). Springer, Cham.

[59] Kohl, S., Bonekamp, D., Schlemmer, H. P., Yaqubi, K., Hohenfellner, M., Hadaschik, B., ... & Maier-Hein, K. (2017). Adversarial networks for the detection of aggressive prostate cancer. arXiv preprint arXiv:1702.08014.

[60] Nie, D., Trullo, R., Lian, J., Petitjean, C., Ruan, S., Wang, Q., & Shen, D. (2017, September). Medical image synthesis with context-aware generative ad-

versarial networks. In International conference on medical image computing and computer-assisted intervention (pp. 417-425). Springer, Cham.

[61] Costa, P., Galdran, A., Meyer, M. I., Abramoff, M. D., Niemeijer, M., Mendonça, A. M., & Campilho, A. (2017). Towards adversarial retinal image synthesis. arXiv preprint arXiv:1701.08974.

[62] Xue, Y., Xu, T., Zhang, H., Long, L. R., & Huang, X. (2018). Segan: Adversarial network with multi-scale l 1 loss for medical image segmentation. Neuroinformatics, 16(3), 383-392.

[63] Chintala, S., Denton, E., Arjovsky, M., & Mathieu, M. (2016). How to train a GAN? Tips and tricks to make GANs work. Github. com.

[64] Patcha, A., & Park, J. M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer networks, 51(12), 3448-3470.

[65] Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks?. arXiv preprint arXiv:1411.1792.

[66] Chuquicusma, M. J., Hussein, S., Burt, J., & Bagci, U. (2018, April). How to fool radiologists with generative adversarial networks? A visual turing test for lung cancer diagnosis. In 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018) (pp. 240-244). IEEE.

[67] Akoglu, L., Tong, H., & Koutra, D. (2015). Graph based anomaly detection and description: a survey. Data mining and knowledge discovery, 29(3), 626-688.

[68] Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2013). Network anomaly detection: methods, systems and tools. Ieee communications surveys & tutorials, 16(1), 303-336.

[69] Lee, W., & Xiang, D. (2000, May). Information-theoretic measures for anomaly detection. In Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001 (pp. 130-143). IEEE.

[70] Brame, C. J. (2018). Active Learning, Vanderbilt University Center for Teaching, 2016. Saatavissa (luettu 4.5. 2019): https://cft. vanderbilt. edu/guides-sub-pages/active-learning.

[71] Felder, R. M., & Brent, R. (2009). Active learning: An introduction. ASQ higher education brief, 2(4), 1-5.

[72] Wolfe, K. (2006). Active learning. Journal of teaching in travel & tourism, 6(1), 77-82.