AN IMPROVED BIDIRECTIONAL GENERATIVE ADVERSARIAL NETWORKS BASED APPROACH FOR ANOMALY DETECTION

(ANORMALLİK TESPİTİNDE ÇİFT YÖNLÜ ÜRETKEN ÇEKİŞMELİ AĞLAR YAKLAŞIMININ GELİŞTİRİLMESİ İÇİN BİR YÖNTEM)

by

MUHAMMET OĞUZ KAPLAN, B.S.

Thesis

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

in

INDUSTRIAL ENGINEERING

in the

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

of

GALATASARAY UNIVERSITY

Supervisor: Assoc. Prof. Dr. Sadettin Emre ALPTEKİN

JULY 2020

This is to certify that the thesis entitled

AN IMPROVED BIDIRECTIONAL GENERATIVE ADVERSARIAL NETWORKS BASED APPROACH FOR ANOMALY DETECTION

prepared by **MUHAMMET OĞUZ KAPLAN** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering Department** at the **Galatasaray University** is approved by the

Examining Committee:

Assoc. Prof. Dr. S. Emre ALPTEKİN
Supervisor, **Industrial Engineering Department**
Galatasaray University ______________

Assoc. Prof. Dr. M. Ebru ANGÜN
Industrial Engineering Department
Galatasaray University ______________

Assoc. Prof. Dr. Umut ASAN
Industrial Engineering Department
Istanbul Technical University ______________

Date: ______________

ACKNOWLEDGMENTS

First and foremost, I would like to thank my thesis advisor Assoc. Prof. Dr. S. Emre Alptekin for his support, guidance and patience through my masters education. His guidance helped me in all the time of research and writing of this thesis.

I have limitless gratitude to my family, to my father Sefer Ali, my mother, Fatma. If I am here in this position today, it is because of them. They all have immeasurable endeavor in every step I took so far. I am so lucky to born in this family.

I would like to show my gratitude to my friends at the Inzva for their support and friendship. I thank Onur Boyar for having invaluably generative and competitive discussions and his beneficial insights. I also thank Ramazan Yarar and Ali Akay for their companionship, friendship, and productive discussions at the Inzva. I also appreciate the Inzva for being an intermediary to bring this incredible team together, and for providing a motivational environment and support.

July 2020
M. Oğuz Kaplan

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Anomaly detection is considered as a challenging task due to its imbalanced and unlabelled nature. Numerous machine learning methods are applicable to the anomaly detection task. Conventional machine learning algorithms, such as supervised anomaly detection methods require labeled data sets and can obtain reasonable achievements on balanced data sets. However, they mostly suffer from the class imbalance problem. Unsupervised anomaly detection methods, on the other hand, assume that the more significant part of the data is normal and are inclined to label the least fit instances as anomalies. Semi-supervised methods create structures from normal data, which represents standard data distribution. To overcome this challenge, the combination of different machine learning approaches such as supervised, unsupervised, semi-supervised learning are proposed in the literature. With the advent of neural networks and generative models, different methodologies derived from neural networks are applied to anomaly detection tasks. In this study, we use the KDDCUP99 and Credit Card Fraud Detection data set, consider them as an anomaly detection task, and implement Bidirectional Generative Adversarial Networks, considering it as a one-class anomaly detection algorithm. Since generator and discriminator are highly dependent on each other in the training phase, to reduce this dependency, in this paper, we propose three different training approaches for Bidirectional Generative Adversarial Networks by adding extra training steps to it. We also demonstrate that proposed approaches increased the performance of Bidirectional Generative Adversarial Networks on anomaly detection task.

Keywords : generative adversarial networks, bidirectional generative adversarial networks, intrusion detection, fraud detection, anomaly detection

ÖZET

Anomali tespiti, anomalilerin yetersiz sayıda bulunması ve etiketlenmemiş olması nedeniyle zorlu bir problem olarak kabul edilir. Anomali tespiti problemi için çok sayıda makine öğrenme yöntemi uygulanabilir. Denetimli anomali tespiti yöntemleri gibi geleneksel makine öğrenme algoritmaları, etiketli veri kümeleri gerektirir ve dengeli veri kümelerinde makul başarılar elde edebilir. Ancak, çoğunlukla sınıf dengesizliği probleminden muzdariptirler. Denetimsiz anomali tespit yöntemleri ise verilerin daha önemli kısımlarının normal olduğunu ve anormallik olarak en az uyuşan durumları etiketlemeye meyillidirler. Yarı denetimli yöntemler, normal verilerden standart veri dağılımını temsil eden yapılar oluşturur. Bu problemin üstesinden gelmek için, literatürde denetimli, denetimsiz, yarı denetimli öğrenme gibi farklı makine öğrenme yaklaşımlarının kombinasyonu önerilmektedir. Yapay sinir ağlarının ve üretken modellerin ortaya çıkmasıyla, sinir ağlarından türetilen farklı metodolojiler anomali tespit görevlerine uygulanmaktadır. Bu çalışmada, KDDCUP99 ve Kredi Kartı Dolandırıcılık Tespiti veri kümelerine, bir anomali tespit problemi olarak yaklaşılmış ve Çift Yönlü Çekişmeli Ağlar tek sınıf bir anomali tespit algoritması olarak değerlendirilerek bu verisetlerine uygulanmıştır. Üretici ağ ve ayırt edici ağ eğitim aşamasında birbirine oldukça bağımlı olduğundan, bu bağımlılığı azaltmak için, bu çalışmada, Çift Yönlü Çekişmeli Ağlar'ın eğitim aşamasına ekstra adımları ekleyerek üç farklı eğitim yaklaşımı önerilmiştir. Ayrıca önerilen yaklaşımların anomali saptama görevinde Çift Yönlü Çekişmeli Ağlar'ın performansını artırdığını gözlemlenmiştir.

Anahtar Kelimeler : çekişmeli üretici ağlar, çift yönlü çekişmeli üretici ağlar, siber saldırı tespiti, dolandırıcılık tespiti, anomali tespiti

LIST OF ABBREVIATIONS

1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
AAE	Adversarial Autoencoders
ALICE	Adversarially Learned Inference with Conditional Entropy
AnoGAN	Anomaly Detection with Generative Adversarial Networks
AnoVAEGAN	Anomaly Detection with Variational Autoencoder + Generative Adversarial Network
AVB	Adversarial Variational Bayes
BiCoGAN	Bidirectional Conditional Generative Adversarial Networks
BiGAN	Bidirectional Generative Adversarial Networks
BIRCH	Balanced Iterative Reducing and Clustering Using Hierarchies
CBLOF	Cluster Based Local Outlier Factor
CGAN	Conditional Generative Adversarial Networks
CLARANS	Clustering Large Applications Based on Randomized Search
CLIQUE	Clustering In Quest
COF	Connectivity Based Outlier Factor
CURE	Clustering Using Representatives
DBSCAN	Density-based Spatial Clustering of Applications with Noise
DCGAN	Deep Convolutional Generative Adversarial Networks
DENCLUE	Density-based Clustering
DiscoGAN	Discover Relations Between Different Domains Generative Adversarial Networks
EGBAD	Efficient Generative Adversarial Networks Based Anomaly Detection
f-AnoGAN	Fast Anomaly Detection with Generative Adversarial Networks
FN	False Negative
FP	False Positive
GAN	Generative Adversarial Networks
HS-Tree	Half Space Tree
INFLO	Influenced Outlierness
JS Divergence	Jensen-Shannon Divergence
KL Divergence	Kullback–Leibler Divergence
kNN	K Nearest Neighbor
LOCI	Local Correlation Integral
LOF	Local Outlier Factor
LS-GAN	Loss Sensitive Generative Adversarial Networks
LSGAN	Least Squares Generative Adversarial Networks
MkNN	Mutual K Nearest Neighbor
ML	Machine Learning

MR	Magnetic Resonance
MSE	Mean Squared Error
OC-SVM	One Class Support Vector Machines
PAC-RPAD	Probability Approximately Correct Rare Pattern Anomaly
PCA	Principle Component Analysis
ReLu	Rectified Linear Activation Unit rectified linear activation unit
RS-Forest	Randomized Space Forest
SAE	Sparse Autoencoder
SMOTE	Synthetic Minority Over-sampling Technique
STING	Statistical Information Grid
SVDD	Support Vector Data Description
TN	True Negative
TP	True Positive
VAE	Variational Autoencoder
VAEGAN	Variational Autoencoder + Generative Adversarial Network
WAVECLUSTER	Wavelet Based Clustering

1 INTRODUCTION

Anomalies are data patterns which do not comply with a clear concept of normal data behavior. The unexpected manners whose pattern significantly different from other observations are named as anomalous observations. Detecting anomalies aims at discovering unexpected objects or occurrences in data sets that differ from the standard.

Categorizing, specifying, and classifying abnormal observations is crucial for data mining, decision-making, and business intelligence. Advances in artificial intelligence, machine learning, and deep learning algorithms have allowed automatic and real-time identification of anomalies. Anomaly detection has many applications in real-life cases such as :

— Intrusion detection

— Fraud detection

— Fault detection

— Detecting healthcare anomalies

— Detecting manufacturing anomalies

— Detecting anomalies in computer vision tasks

— Detecting anomalies in text data

Numerous machine learning methods are applicable to the anomaly detection task. Conventional machine learning algorithms, such as supervised anomaly detection methods require labeled data sets and can obtain reasonable achievements on balanced data sets. However, they mostly suffer from the class imbalance problem. Unsupervised anomaly detection methods, on the other hand, assume that the more significant part of the data is normal and are inclined to label the least fit instances as anomalies. Semi-supervised methods create structures from normal data, which represents standard data distribution. Anomaly detection, due to its imbalanced or unlabelled nature, may require a combination of the machine learning approaches mentioned above. Other challenges on anomaly detection tasks described by the study of (Chandola et al., 2009) as follows :

— Since, in many cases, anomaly observations and normal data are not separable accurately, it is hard to describe the boundaries of normal data. In these cases, anomalies might be misclassified as normal observation and vice-versa

— Commonly, malicious actions are tried to conform to normal data behavior. This similarity makes anomaly detection tasks harder in these cases.

— Due to the transformation in many businesses, the concept of normal observations keeps on changing, and today's normal behavior can not be an efficient representative of normal data characteristics in the future.

— Different domains have different dynamics and work on distinct scales. Therefore, it is not an efficient way to apply one technique established in one domain to another.

— Acquiring labeled data for training and test set sufficiently is another main difficulty.

— In real-life cases, data noise makes the process of distinguishing normal data from anomalous data more difficult.

One of the most challenging problems of anomaly detection is the class imbalance problem. In a data set, the amount of observations in each class is described as class distribution. When the class distribution in the training set is not balanced, the class imbalance problem occurs. There are two fundamental causes of class imbalance problem, namely properties of the domain and sampling. When the cause of the class imbalance problem is the sampling error, the imbalance problem can be corrected by using advanced sampling methods.

In some real cases, the natural occurrence of any class may dominate other classes. In such cases, gathering more observations from the minority class's domain to enhance the class distribution is impracticable. Alternatively, an advanced model is needed to learn to distinguish each class from each other. In this study, we are mostly interested in the imbalance problem caused by the properties of the domain.

If class distribution in training data is a little skewed, this is called a slight imbalance. On the other hand, if hundreds or thousands of times more observations from one class are compared to the other class, this classification problem is named a severe imbalance. While slight imbalance is not considered as a problem and standard machine learning classification approaches can be built on the data with a slight imbalance, to build a model on data with severe imbalance problems can be seen as a difficult task and may require specialized techniques. On the imbalanced classification task, predicting minority class correctly is usually the primary concern.

Most ML classification algorithms are modeled and illustrated on the data with the balanced class distribution. This implies that the model can mislead by the imbalanced class distribution and neglects the observation from minority class whose prediction is the more valuable and primary concern of the client. In this thesis, we focused on solving the severe class imbalanced problem with one-class generative adversarial networks approaches and extended Bidirectional Generative Adversarial Networks (BiGAN).

With the advent of neural networks, different methodologies derived from neural networks have been applied to anomaly detection to overcome this challenging task. Autoencoders performed very well and contributed to anomaly detection methodologies. By adding variational inference to neural networks, probabilistic approaches have been introduced, and variational autoencoders (Kingma and Welling, 2013) have been utilized to conduct anomaly detection task more principled by using reconstruction probability rather than reconstruction error (An and Cho, 2015).

After (Goodfellow et al., 2014) introduced Generative Adversarial Networks, GANs have emerged as a leading technique for both unsupervised and semi-supervised learning on anomaly detection problem. GANs' architecture is evolved in such a way that it has been used to produce an anomaly score. (Schlegl et al., 2017) speculated that the representative vector of data in the latent space of a GAN stands for the real distribution of the data. Intuitively, well trained GANs can learn the input data distribution, reproduce data from latent representation in corresponding distribution, and distinguish the instances from corresponding distribution.

Two deep neural networks, the generator and discriminator, constitute GANs' structure. Generators and discriminators are trained in turn by playing an adversarial game. Discriminator takes the output of the generator (fake data) and real data as input. The discriminator aims to distinguish fake data from real data. The generator focuses on convincing the discriminator so that the generator can learn to generate realistic data.

Generators and discriminators are highly dependent on each other in the training phase. This dependency creates a hard problem for the generator's training. In this thesis, to reduce this dependency and force generator produce more reliable data that represents the real data, we added extra training step for generator independent from the discriminator, which updates only the generator's parameters by calculating the loss between the input of the encoder and the output of the generator at the end of each epoch. We proposed three different approaches. Our contribution is that these approaches showed that updating the generator independent from discriminator improved the generator performance, thus discriminator performance, and we increased the performance of Bi-GAN on anomaly detection task. The proposed approach is explained in detail in the section of 3.2.3.

The rest of this thesis is organized as follows. Section 2 summarizes the literature regarding this thesis's topic. The methods studied in this thesis are presented in section 3. Section 4 demonstrates the numerical results on the test data. Section 5 discusses future studies and gives concluding results.

2 LITERATURE REVIEW

2.1 Overwiev

Anomaly detection has been the topic of numerous surveys, reviews and researches. In this section, the related literature regarding the topic of this thesis is reviewed. Anomaly detection approaches are mentioned in five main streams. Proximity-based algorithms, clustering-based algorithms, tree-based algorithms, kernel-based algorithms and deep approaches for anomaly detection are reviewed in sections 2.2, 2.3, 2.4, 2.5 and 2.6, respectively. The contribution and motivation of this thesis are presented in section 2.7. In figure 2.1, a summary chart of anomaly detection algorithms mentioned in this chapter are illustrated.

2.2 Proximity Based Algorithms

The primary motivation behind proximity-based anomaly detection algorithms is to reveal the neighborhood relationship. This relationship is determined by the distance or derivative of distance-based methods (Knox and Ng, 1998; Ramaswamy et al., 2000; Angiulli and Pizzuti, 2002)

(Brito et al., 1997) studied on mutual-nearest-neighbor graph for detecting anomalies in multivariate data sets and introduced Mutual-Nearest Neighbor (*MkNN*) graph-based undirected graph method, which is a specified type of *kNN*. *MkNN* graphs consist of edges between vectors in the case that vectors are k-neighborhoods of each other. Each node has a pointer to its k nearest neighbors. Anomalies are assumed to be a component connected only to one vector.

(Knox and Ng, 1998) focused on detecting distance-based anomaly detection methods and proposed optimized cell-based algorithm and two basic algorithms that they have both complexity of $O(kN^2)$ where k represents the dimensionality of data and N stands for the observation in the corresponding data set. While the cell-based algorithm is acceptable when $k <= 4$, nested-loop algortihm performs well in the cased that $k >= 5$. While taking advantage of being simple, (Knox and Ng, 1998) had certain drawbacks that the distance is required as input and dependence between complexity and dimensionality are linear. (Ramaswamy et al., 2000) concentrate on the shortcomings of (Knox and Ng, 1998) and introducing a new description for novelties. (Ramaswamy et al., 2000) proposed a partition-based method for detecting outliers that takes the distance to kth nearest neighbor as anomaly score measurement. The proposed method first divides the data into clusters, and pruning is applied to clusters to obtain intense clusters. On the final step, the distance to each point in the partition is calculated. The proposed algorithm performance exceeded the nested-loop and index-based algorithms' performances on not only the area of detecting anomalies but also on the scalability.

(Angiulli and Pizzuti, 2002) recommended a new method to overcome the detecting anomalies on high dimensional data sets. They utilized weighted distances. Weights are calculated as the sum of each observations' distance to k nearest neighbors. Find the k nearest neighbors of each observation is optimized by using the Hilbert space-filling curve. Anomaly score of each point is defined as the weights calculated for each observation. (Angiulli and Pizzuti, 2002) concluded that increasing the dimensionality by fixing the size of data decreased the number of iterations required by the proposed method to obtain a solution.

(Hautamaki et al., 2004) proposed a method of detecting anomalies using indegree numbers that make use of k-nearest neighbor graphs. Observations connected to at most 5 neighborhoods are defined as an anomaly. In contrast to (Brito et al., 1997),in this study kNN graph is taken into consideration as weighted directed graph. Anomaly classification task conducted according to the in-degree number in the graph.(Hautamaki et al., 2004) performed much better than (Knox and Ng, 1998) on real data sets.

On clustering algorithms, observations do not belong to any cluster is called anomaly or noise, and the performances of clustering algorithms are highly dependent on model parameters. (Breunig et al., 2000) proposed a new algorithm for detecting anomalies that does not need the concept of clustering on multivariate data sets to overcome the drawbacks of clustering algorithms on anomaly detection tasks. The local outlier factor is based on a local density concept, where locality is given by k nearest neighbors, whose distance is used to estimate the density. Observations with a significantly lower density than their neighbors are determined as an anomaly.

The effectiveness of density-based approaches will decrease if the density of anomaly observation is close to the density of its neighbors. (Tang et al., 2002) proposed a connectivity-based outlier factor (COF) algorithm, which deals with the case that anomaly has a similar pattern as the neighborhoods have. The COF structure improved the effectiveness of the local outlier factor (LOF). However, the certainty of the generated anomaly score will reduce when the groups from different densities are exceedingly close to each other. The drawback mentioned above is later solved in the Influenced Outlierness (INFLO) algorithm which is introduced by (Jin et al., 2006)

(Papadimitriou et al., 2003) introduced a method of $Local\ Correlation\ Integral$ and $multi-granularity\ deviation\ factor$. Additionally proposed $aLOCI$, which uses the approximation of box-counting to reduce complexity to $O(kN)$ and fastens the anomaly detection process to deal with large scale data sets. It ıs proven that the proposed method is successful in dealing with both local density and multiple granularities.

(Kriegel et al., 2009) formulate a new local density-based, probability scoring method for anomaly detection tasks combining LOF (Breunig et al., 2000) and LOCI (Papadimitriou et al., 2003). The algorithm 's based on the assumption that distance between instances acts like the positive leg of gaussian distribution. Generating an anomaly score in the range of [0,1] as the probability contributes to the interpretability and comparison of observations in the data set.

2.3 Clustering Based Algorithms

Clustering is a Machine Learning technique that involves the grouping of data points. Given a set of data points, a clustering algorithm can be used to classify each data point into a specific group. In theory, data points that are in the same group should have similar properties or features, while data points in different groups should have profoundly different properties or features. Clustering is a method of unsupervised learning and is a common technique for statistical data analysis used in many fields.

Clustering-based techniques depend on the following assumptions. Firstly, while normal data belongs to a cluster, anomalous observations is not a member of any cluster. Secondly, normal observations are members of dense clusters ; on the other hand, anomalous observations belong to small clusters.

K-means, which is proposed by James MacQueen, is the perhaps most popular and simplest clustering algorithm (MacQueen et al., 1967). K-means algorithm groups the observations according to the given number of clusters by initializing the centroids randomly and updating the centroids in each iteration. If the data points do not have formal shape, K-means can be ineffective in grouping the observations. Moreover, outliers in the training data can probably influence the centroids of clusters in each iteration. This may increase the misclassification error of observation in the test set since the outlier in the training set increases the distance threshold. Lastly, there is no restriction to determine the number of observations in clusters. For the reasons mentioned above, the K-means algorithm does not give good results most of the time.

(He et al., 2003) proposed a clustering-based method that utilizes k-means, a Cluster-Based Local Outlier Factor (CBLOF). The anomaly score of the observation is obtained by measuring the distance to the next populous cluster. The main drawbacks of this method are the hardship of choosing the correct number of clusters and the impracticability of reproducing just the same outlier score since the clustering algorithms are non-deterministic.

Clustering algorithms like CLARANS (Ng and Han, n.d.), DBSCAN (Ester et al., 1996), BIRCH (Zhang et al., 1996), CURE (Guha et al., 1998), STING (Wang et al., 1997), WAVECLUSTER (Sheikholeslami et al., 1998), DENCLUE (Hinneburg et al., 1998), CLIQUE (Agrawal et al., 1998) are slightly able to detect anomalies. Clustering is not ideal for all the problems related to anomaly detection, but combining clustering techniques with others like smart feature extraction can help to solve many problems. Commonly, clustering algorithms ignore anomalies in the process of grouping observations, which leads to non-optimal solutions on anomaly detection tasks. Since they are designed to group the observations in an optimized way, outlier detection is not their first concern, and they do not provide any measure about the outlying observations.

2.4 Tree Based Algorithms

In tree-based anomaly detection methods, an anomaly score is assigned to each observation, and each observation is ranked and ordered hierarchically according to their anomaly score. In tree-based anomaly detection algorithms, each node keeps weight to calculate the anomaly score. The anomaly score is calculated with the combination of weights on the path of the corresponding observation pursues. What sets a specific tree-based anomaly detection algorithm to other tree-based algorithms is the different

Table 2.1: Node weight of tree based algorithms

Algorithm	Internal Node Weight	Leaf Node Weight
Isolation Forrest	-1	-1
HS-Trees	0	(Tan et al., 2011)
RS-Forrest	0	(Wu et al., 2014)
RPAD	normalized pattern freq.	normalized pattern freq.
Random Projection Forrest	log-prob. at the node	log-prob. at the node

weight choice approaches showed on table 2.1.

(Liu et al., 2008) proposed the Isolation Forest algorithm that aims to separate outliers rather than characterize normal observations and has a linear time complexity with a low constant and memory usage. Isolation trees with the same architecture of Binary Search Tree constitute the Isolation Forrest. These trees are obtained with random partitioning at each node. While conducting random partitioning, the features and thresholds are chosen uniformly random. Observations with shorter than average path length are labeled as an anomaly in this method. The algorithm has two hyperparameters : number of trees and sub-sampling size. In contrast to other machine learning algorithms, using small sub-samples is expected to create more efficient isolation trees since swamping and masking are lessened.

Conducting an anomaly detection task in streaming data has the following aspects. First, memory problems occur while keeping the whole stream for offline learning algorithms since the stream is infinite. Second, usually, classifiers suffer from insufficiency of anomalous observations in stream data. Third, since the stream data changes the characteristics in the progress of time, the trained model must stay focused on various detail of stream. (Tan et al., 2011) proposed a method of Streaming Half-Space Trees for conducting anomaly detection on stream data, which concerns the problem mentioned above. Instead of producing trees from training data, HS-Tree's, dissimilar to other decision trees, utilize only data space dimensions to produce its tree structure without requiring no split point and feature calculation. HS-Tree is an effective algorithm in imbalance anomaly detection tasks since it is one class detector with $O(1)$ space and time complexity and robust to emerging data streams.

In 2014, to overcome the difficulty mentioned above of the anomaly detection on stream data, (Wu et al., 2014) introduced a one-class semi-supervised algorithm RS-Forest. The proposed algorithm is based on estimating density by use of RS-Trees. At the beginning of the algorithm, trees are initialized randomly without data, and statistical evaluations are conducted to estimate the potential direction of attribute ranges to make the use of it while splitting the tree. The aim of building trees in this way is to ensure to be robust in evolving characteristic of stream data and to augment the diversification of ensembles that boosts to have better density estimation. As in the (Tan et al., 2011), this method is one pass resulting in linear time and constant space complexity, and since the algorithm is one class, it does not suffer from an imbalanced class problem.

In PAC learning theory, characterizing the concept of a hypothesis space, computing the relationship between the complexity of this space and the volume of training data needed to establish an acceptable hypothesis in the space and relaxing the limit of identifying the most fitting hypothesis are the main steps.(Siddiqui et al., 2016) proposed Probability Approximately Correct Rare Pattern Anomaly Detection (PAC-RPAD) framework for discovering the unusual patterns by following the same procedure above. Observations that meet lower probability patterns are determined as anomalies. Outlier detection benchmarking researches (Emmott et al., 2013) illustrated that algorithms focused on finding rare patterns for detecting anomalies, such as RPAD-style approach, inclined to outperform the common pattern approaches such as OC-SVM (Schölkopf et al., 2001). Additionally, empirical results show that PAC-RPAD is competitive with the Isolation Forrest (Liu et al., 2008) when the same pattern space is utilized and provides fast convergence.

(Chen et al., 2015) introduced a novel anomaly detection algorithm with an online updatable structure to get the better of new outliers. Random Projection Forest. The splitting rule of Random Projection Forest is adjusted to maximize the marginal space for anomalies. A decision tree carries out hierarchical clustering, and the elements are dissociated iteratively as right and left child nodes. On the other hand, in (Chen et al., 2015), each observation are projected into an optimal direction derived from a randomly determined set. The KL Divergence is utilized to check the fitting quality of the experimental density of projected observations. To obtain optimal splits, this divergence is maximized. In test, the aggregated density of each observation is calculated to analyze the irregularity by comparing it with the determined threshold value. This architecture enables the model to be robust to under or overfitting problems.

2.5 Kernel Based Anomaly Detection

Perhaps the most famous example of kernel-based one classification algorithm is One-Class Support Vector Machines (OC-SVM) (Schölkopf et al., 2001). The main motivation of this study arises from the study of (Vapnik, 1963). (Vapnik, 1963; Vapnik and Chervonenkis, 1974) proposed an algorithm that classifies the group of unlabelled observations by utilizing a hyperplane that distinguishes the points from the origin. However, the proposed algorithm was restricted with the linear decision rules, and coping with the anomalies was impossible. (Schölkopf et al., 2001) utilized two tricks to deal with above-mentioned drawbacks of Vapnik : Kernel trick and v trick (Schölkopf et al., 2000). While kernel trick enabled (Schölkopf et al., 2001) to represent the data in high dimensional space to enhance the possibility of distinguishing from the origin, v trick contributed to the integration of prior belief on the portion of anomalies into the model.

(Tax and Duin, 2004), motivated by the Support Vector Classifier, proposed a new method for detecting anomalies on a multidimensional dataset by drawing a spherically shaped boundary over the data set and avoiding to approximate a density to the data set. They illustrated that altering the ball shaped boundary to a flexible boundary contributed to the control of anomaly sensitivity. While SVDD with polynomial kernel had the substantial impact of the norm of the object vectors, in opposition to Support Vector Classifier, SVDD with Gaussian kernel achieved competitive descriptions.

The Support Vector Data Description and one Class Support Vector Machines are analogous. Support vector data description (Tax and Duin, 2004) finds the smallest hypersphere that contains all samples, except for some outliers. One-class SVM (OC-SVM) separates the inliers from the outliers by finding a hyperplane of maximal distance from the origin. SVDD and OC-SVM are also equivalent in the case that all samples lie on a hypersphere centered at the origin, and are linearly separable from it.

2.6 Deep Approaches for Anomaly Detection

With the advent of neural networks, autoencoders, variational autoencoders, GANs have been widely used thanks to their training architecture. Autoencoders, with their reconstruction loss as an anomaly score, is used in anomaly detection task (Sabokrou et al., 2016; Zhou and Paffenroth, 2017). Obtaining the representation of the distribution of normal data enables us to generate normal-like data, similarly enables us to detect the data from a different distribution.(An and Cho, 2015) proposed reconstruction probability on anomaly detection tasks instead of reconstruction cost and used the generative structure to obtain the anomaly score.

Considering the insufficient computational scalability and curse of dimensionality, traditional anomaly detection algorithms such as One-Class SVM (Schölkopf et al., 2001) and Kernel Density Estimation (Parzen, 1962) generally do not succeed in the high dimensional data. Autoencoders are not aimed to detect anomalies on given data set directly but utilized for dimensionality reduction. The main struggle in autoencoders is choosing the most suitable compactness of data representation degree because of the unsupervised nature of autoencoders. Motivated by the kernel-based one-class classification and minimum volume estimation, (Ruff et al., 2018) proposed the Deep Support Vector Data Description (Deep SVDD) algorithm that trains neural network by minimizing the volume of hypersphere to extract the common factors of variation. By doing so, the Deep SVDD algorithm covers the compactness of representation.

Generative models, recently, are gaining popularity in detecting anomalies while conducting anomaly detection tasks. Since anomalies are rarely occurring in data, unsupervised models perform very well when compared to conventional supervised machine learning algorithms. In various network intrusion, fraud detection, medical domain, computer vision tasks, these models achieved significant results (Zenati et al., 2018; Schlegl et al., 2017; Bian et al., 2019).

The main intuition behind using GAN on detecting novelties is that generator is taught to reproduce normal data and discriminator is used to discriminate normal and non-normal data. As an unsupervised learning algorithm, generative adversarial networks are used to learn the representation of the distribution of normal data and produce anomaly scores according to the latent representation of each instance (Schlegl et al., 2017). To obtain the anomaly score, residual loss between generated data and test instance is combined with the discriminator loss. Discriminator loss is obtained by feeding the discriminator network with the generator's output (Schlegl et al., 2017). (Zenati et al., 2018) used Bidirectional GAN (Donahue et al., 2016) architecture that consists of encoder, generator and discriminator networks. Encoder and generators outputs and inputs together feed the discriminator. While encoder networks are trained to obtain the latent representation of input instances, generator networks are trained to produce normal data from given noise input.

Similar to (Schlegl et al., 2017), the anomaly score is calculated with residual loss of encoder and generator outputs and added to discriminator loss.(Akcay et al., 2018) proposed a different pipeline that combines autoencoder network as the generator and another encoder network with discriminator. Due to (Akcay et al., 2018), architecture, contextual, encoder, and adversarial loss constitutes the anomaly score. (Chen et al., 2018) tackled the fraud detection problem as a one-class anomaly detection task and combined sparse autoencoders and GAN to detect fraudulent activities. The proposed architecture is formed of two steps. First, SAE is trained separately from GAN to obtain critical features of the observations. After extracting the key features of the data, the GAN algorithm is trained with the extracted features from SAE. Dissimilar to (Schlegl et al., 2017; Zenati et al., 2018), they only used discriminator component of GAN to distinguish the fraudulent observations. SAE + GAN outperformed to the SVDD and One-Class GP.

(Fiore et al., 2019) overcome the imbalance dataset problem by using GAN as an oversampling technique for minority class, instead of anomaly classifier. They trained a classifier with augmented data by GAN and original data. They concluded that classifiers trained on augmented data perform better than the classifier performed on the original data. They also compared their approach with the SMOTE, which is a similar approach for oversampling. While lower sensitivity is usually obtained the data augmented by SMOTE, specificity is inclined to be more when compared to the proposed approach.

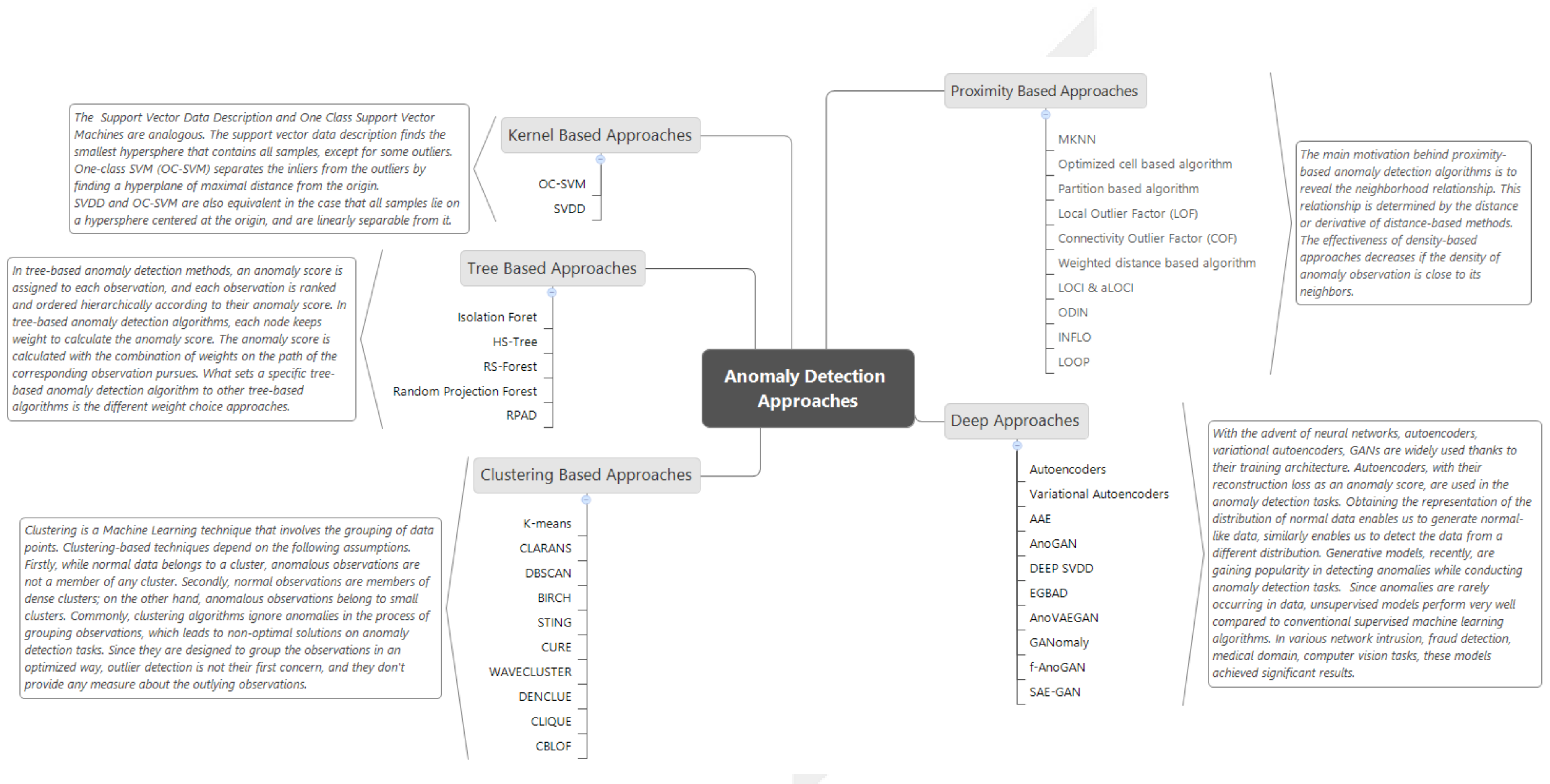
In the medical domain, (Baur et al., 2018) dealt with the outlier segmentation of brain MR images. (Baur et al., 2018) proposed anoVAEGAN with the adaptation of VAE-GAN (Donahue et al., 2016) to anomaly detection task. In this study, L_1 loss between the input image and the reconstructed image is used as an anomaly score. (Chen and Konukoglu, 2018) focused on discovering brain lesions by using constrained adversarial autoencoders. AAE (Makhzani et al., 2015) implements an encoding-decoding architecture equivalent to VAE, but instead of KL divergence uses Jensen-Shannon(JS) divergence on adversarial training. (Chen and Konukoglu, 2018) proposed two-step anomaly detection structure. On the first step, autoencoder learns a latent representation of health images, and on the second step, non-normal images are given as input.

Since non-normal images are less likely to be reconstructed, L_2 loss is calculated as an anomaly score. (Schlegl et al., 2019) proposed f-AnoGAN, which is aimed to detect anomalous image segments and images, especially in the medical domain. What sets f-AnoGan apart from anoGAN (Schlegl et al., 2017) is that f-anoGAN utilizes improved Wasserstein GAN (Arjovsky et al., 2017a) architecture instead of DCGAN (Radford et al., 2015)and by replacing iterative gradient descent with learned mapping, it accelerates the process of mapping input to latent space. The training phase has two steps : GAN training and encoder training dependent on the GAN model. While the f-AnoGAN method is suitable for detecting anomalous 1D, 2D, 3D images, it has some limitations on locating anomaly in the image and can be considered as coarse localization.

2.7 Motivation and Contribution

The impact of a generator weight depends on the impact of the discriminator weights it feeds into. So backpropagation starts at the output and flows back through the discriminator into the generator. At the same time, we do not want the discriminator to change during generator training. Training generator with the error of discriminator makes the process deeper in terms of generator. This thesis is focused on reducing the effect of this dependency and on improving the generator' learning process and dependently discriminator's learning process, and aims to obtain improvements on the test set performances.

In this study, we proposed three different training structures for BiGAN by adding one more step for the generator's training phase. Similar to our approaches DiscoGAN (Kim et al., 2017) and DualGAN (Yi et al., 2017) proposed to add additional reconstruction loss term to GAN and ALICE(Li et al., 2017) proposed to add additional reconstruction loss term to BiGAN but these three methods dissociate from us in terms of the model architecture and reconstruction loss. Those approaches also aimed to improve the reconstruction quality of GAN and BiGAN, and they are used in computer vision tasks. In this study, we will apply the BiGAN structure in the anomaly detection task. In addition to BiGAN's training process, we proposed three separate methods for BiGAN's training. In the first method, the generator is trained according to its custom loss at the end of each epoch, and in the second method in addition to the first method, we started BiGAN's training with the pre-trained generator to improve the generation quality of generator consequently discriminators' capability, to obtain more composite anomaly scores, since we use both discriminator and generator to obtain anomaly score for the test set. In the third approach, we only pre-trained generator before BiGAN's simultaneous training begins. We followed the same approach to obtain the anomaly score as in the EGBAD(Zenati et al., 2018).



The Support Vector Data Description and One Class Support Vector Machines are analogous. The support vector data description finds the smallest hypersphere that contains all samples, except for some outliers. One-class SVM (OC-SVM) separates the inliers from the outliers by finding a hyperplane of maximal distance from the origin.
SVDD and OC-SVM are also equivalent in the case that all samples lie on a hypersphere centered at the origin, and are linearly separable from it.

In tree-based anomaly detection methods, an anomaly score is assigned to each observation, and each observation is ranked and ordered hierarchically according to their anomaly score. In tree-based anomaly detection algorithms, each node keeps weight to calculate the anomaly score. The anomaly score is calculated with the combination of weights on the path of the corresponding observation pursues. What sets a specific tree-based anomaly detection algorithm to other tree-based algorithms is the different weight choice approaches.

Clustering is a Machine Learning technique that involves the grouping of data points. Clustering-based techniques depend on the following assumptions. Firstly, while normal data belongs to a cluster, anomalous observations are not a member of any cluster. Secondly, normal observations are members of dense clusters; on the other hand, anomalous observations belong to small clusters. Commonly, clustering algorithms ignore anomalies in the process of grouping observations, which leads to non-optimal solutions on anomaly detection tasks. Since they are designed to group the observations in an optimized way, outlier detection is not their first concern, and they don't provide any measure about the outlying observations.

The main motivation behind proximity-based anomaly detection algorithms is to reveal the neighborhood relationship. This relationship is determined by the distance or derivative of distance-based methods. The effectiveness of density-based approaches decreases if the density of anomaly observation is close to its neighbors.

With the advent of neural networks, autoencoders, variational autoencoders, GANs are widely used thanks to their training architecture. Autoencoders, with their reconstruction loss as an anomaly score, are used in the anomaly detection tasks. Obtaining the representation of the distribution of normal data enables us to generate normal-like data, similarly enables us to detect the data from a different distribution. Generative models, recently, are gaining popularity in detecting anomalies while conducting anomaly detection tasks. Since anomalies are rarely occurring in data, unsupervised models perform very well compared to conventional supervised machine learning algorithms. In various network intrusion, fraud detection, medical domain, computer vision tasks, these models achieved significant results.

Figure 2.1: Anomaly detection approaches summary chart.

3 METHODOLOGY

3.1 Generative Adversarial Networks

3.1.1 Overview

The GAN (Goodfellow et al., 2014) was introduced in 2014. Two main parts constitute the GANs architecture, namely generator and discriminator. Both generator and discriminator are deep neural networks, including the convolutional layer and the dense layer.

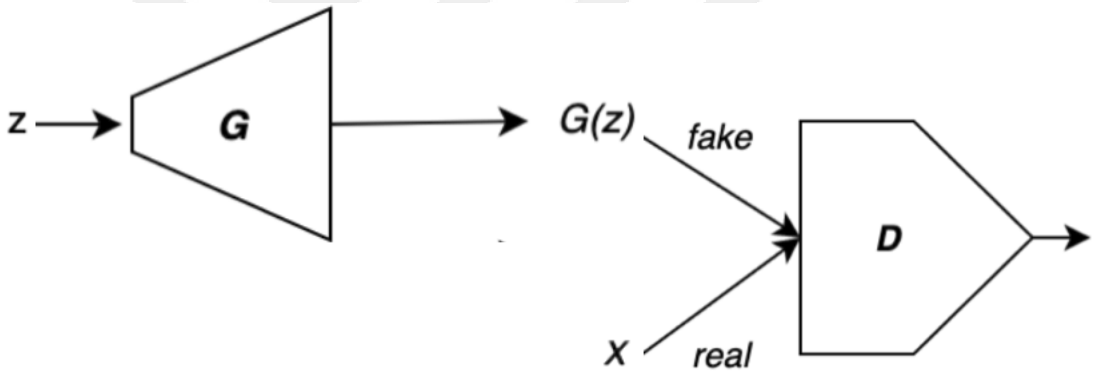


Figure 3.1: GAN architecture. G takes the input z and generates fake x and D is fed by fake x and real x to discriminate true from fake

While the generator takes the input z which is low dimensional and sampled from assumed probability distribution $p(x)$ (e.g., multivariate Gaussian distribution) and generates fake x by mapping the z to certain data dimensionality, the discriminator takes $fake$ and $real$ data as input. The discriminator is expected to distinguish real and fake data points in the training phase of the discriminator.

$$\min_{G}\max_{D} = E_{x\sim p_{data}(x)}[logD(x)] + E_{z\sim p_z(z)}[log(1-D(G(z)))] \qquad (3.1)$$

Generator and discriminator are trained separately playing deception game. While discriminator tries to distinguish fake data generated from noise by generator and real data, the generator aims to convince the discriminator so that generator can learn to

Algorithm 1: GAN algorithm

```
for number of training iterations do
    for k steps do
        Sample $z^t \sim p(z)$;
        Choose $x^t \in X$;
        Update $D(x)$ and $D(G(z))$ by maximizing Eq. 3.1 do not update G;
    end
    Sample $z^t \sim p(z)$;
    Update $G(z)$ by minimizing Eq. 3.1 don't update $D$;
end
```

generate realistic data. With the feedback of discriminator, generator performances on generating realistic data are improved.

Equation 3.1, where x is the real data and $G(Z)$ is the fake data, is maximized by discriminator and minimized by the generator. If p_f is defined as the density of the data generated by the generator and p_{data} is defined as the density or *real* data, the training of GAN is equivalent to process of minimizing KL-divergence between p_{data} and p_f.

There are many applications of GAN especially in computer vision. Some examples are (Zhu et al., 2017a; Dong et al., 2017) for image to image translation, (Ledig et al., 2017) for increasing the image resolution, (Yeh et al., 2017) for image inpainting, (Luc et al., 2016) for image segmentation, (Zhang et al., 2017; Reed et al., 2016) for text to image synthesis. Apart from computer vision, GANs and iys variants are gained popularity in anomaly detection task due to their one-class oriented architecture on which this thesis mainly focuses (Schlegl et al., 2017; Baur et al., 2018; Akcay et al., 2018; Schlegl et al., 2019; Chen et al., 2018; Fiore et al., 2019; Zenati et al., 2018).

3.1.2 Training GAN

Theoretically, GANs are built on non-cooperative minimax game. When one player wins the other loses. While one player aims to minimize the objective, others make an effort to maximize it. Since this is a minimax game, convergence occurs when the discriminator and generator reach the Nash equilibrium. Nash equilibrium is the optimum point of the minimax game from the point of game theory view and reached when one of the competitors stops updating itself in any case. Hence, GANs are hard to train for several reasons for non-convergence.

GANs target to generate a number of fake data from random input, which imitates the real data from different distributions. Doubtless, in the GAN training phase, the most frequent problem is *mode collapse*. Mode collapse is that the generator component of GAN learns to generate only one or a restricted diversity rather than all modes. The mode collapse problem arises when the discriminator is stuck in the local minima, and the generator keeps reproducing the same output since the discriminator fooled easily by the mode generated by the generator. Using different losses like Wasserstein loss is recommended to avoid this situation (Arjovsky et al., 2017b). Unrolled GANs (Metz et al., 2016) is another approach to prevent this situation. To avoid being over-optimized for a unique discriminator, Unrolled GANs utilizes not only the current discriminator but also the output of future discriminator versions for updating the generator component, which is computationally expensive. Applying penalty term for the missing modes in the training phase is presented in the mode regularized GAN (Che et al., 2016). In the literature, there are numerous researches regarding mode collapse problem (Arjovsky et al., 2017b; Salimans et al., 2016; Warde-Farley and Bengio, 2016; Qi, 2019; Mirza and Osindero, 2014)

During backpropagation, the chain rule of differentiation is utilized, and this created a multiplying impact, flowing the gradient from the last layer to the first layer. It gets smaller while flowing to the first layer and sometimes becomes so small that preliminary layers can stop the learning process resulting in vanishing gradients. Using $ReLU$, $LeakyReLU$ instead of *sigmoid* or *tanh* is common and well known approach to avoid vanishing gradient. Wasserstein loss is also formed to avoid the same problem. In the original GAN paper (Goodfellow et al., 2014), proposed modified minimax loss for vanishing gradient problem.

To stabilize the GAN training, various approaches are using different loss functions. Least square GAN (LSGAN) (Mao et al., 2017) focused on improving the generated image quality and implemented the least square loss function for the discriminator. Loss Sensitive GAN (LS-GAN) (Qi, 2019) not only uses Wasserstein loss but also takes advantage of the prior on the real data with utilizing Lipschitz regularity condition.

To stabilize GAN training, an autoencoder like architecture for the discriminator and various energy functions are introduced in Energy-Based GAN (Zhao et al., 2016).

3.2 BiGAN

3.2.1 Overview

While GAN has the ability to generate x from any given z, it does not contains an inverse mapper to generate z for any given x. BiGAN model is introduced by (Donahue et al., 2016) and contains an encoder part that enables the model to map x to z. Dissimilar to proper GAN, discriminator in BiGAN sees not only x but also z. The encoder is also a deep neural network, generally structured as the inverse of the generator.

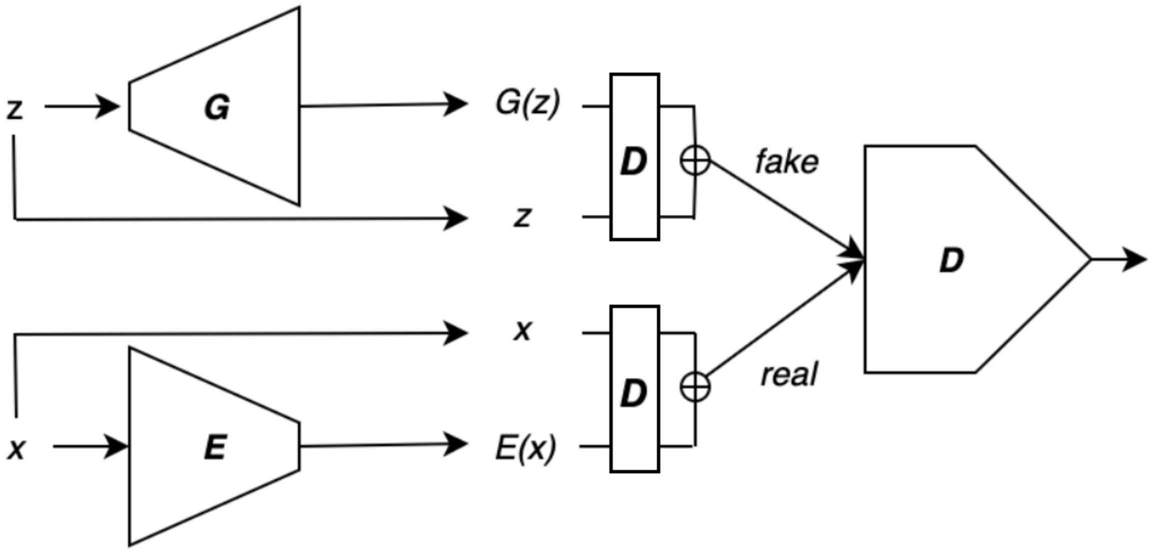


Figure 3.2: EGBAD (Zenati et al., 2018) BiGAN architecture. Discriminator takes the concatenated pair (shown as '$\oplus$') as input. Before concatenation, both inputs are mapped to the same dimension.

BiGAN is an extended version of GAN that adds the encoder part, which maps the input into a latent space. The discriminator takes the concatenated pair z, x as input. Differing from standard GAN, in BiGAN, discriminator takes not only input but also latent representations (generator's input and encoder's output) into account. Loss function is as follows (Donahue et al., 2016) :

$$\min_{G,E}\max_{D} = -E_{x\sim p_x}[E_{z\sim p_E(\cdot|x)}[logD(x,z)]] + E_{z\sim p_z}[E_{x\sim p_G(\cdot|z)}[log(1-D(x,z))]] \quad (3.2)$$

Algorithm 2: BiGAN algorithm

```
for number of training iterations do
    for k steps do
        Sample $z^t \sim p(z)$;
        Choose $x^t \in X$;
        $\hat{z} = E(x^t)$;
        $\hat{x} = G(z^t)$;
        Update $D(x, \hat{z})$ and $D(\hat{x}, z)$ by maximizing Eq. 3.2;
    end
    Sample $z^t \sim p(z)$;
    $\hat{x} = G(z^t)$;
    Update $G(z)$ and $E(\hat{x})$ by minimizing Eq. 3.2;
end
```

(Zenati et al., 2018) used BiGAN (Donahue et al., 2016) architecture that consists of encoder, generator and discriminator networks. Encoder and generators outputs and inputs together feed the discriminator. While encoder networks are trained to obtain the latent representation of input instances, generator networks are trained to produce normal data from given noise input. Similar to (Schlegl et al., 2017), the anomaly score is calculated with residual loss of encoder and generator outputs and added to discriminator loss. To obtain anomaly score, equation (3.3) is used as proposed by (Zenati et al., 2018) :

$$A(x) = \alpha L_G(x) + (1 - \alpha) L_D(x) \tag{3.3}$$

$$L_G(x) = \|x - G(E(x))\|_1 \tag{3.4}$$

$$L_D(x) = \sigma(D(x, E(x)), 1) \tag{3.5}$$

3.2.2 Variants of BiGAN

In the literature, various approaches are using BiGAN like structure. They usually focus on image processing and computer vision topics. Some approaches using BiGAN like structure become prominent on improving the learning and generation performance of BiGAN.

Invertible CGAN (Perarnau et al., 2016) used two encoders and assigned the encoders separately to produce a latent representation of example and latent representation of attributes. The generator is fed with the output of the encoders. They illustrated that using two separate encoders end up with better results. In the invertible CGAN approach, pre-trained CGAN is utilized. However, in BiCoGAN (Jaiswal et al., 2018), almost identical to invertible CGAN, encoder, discriminator and generator are trained at the same time, and BiCoGAN put forward an improved the learning performance. Triangle GAN (Gan et al., 2017) used two discriminators and two generators. While the generator aims to learn bidirectional mappings between two domains, discriminators targeted at distinguishing the different sorts of fake data produced by two separate domains. Adversarial Variational Bayes (AVB) (Mescheder et al., 2017) applies variational training to its BiGAN like structure. In CycleGAN (Zhu et al., 2017b), different from the algorithms uses BiGAN like structure, inverse mapping is utilized for mapping source image to the target image, and cycle consistency loss is proposed to keep the balance of training.

3.2.3 Proposed training architecture for BiGAN

In GAN and BiGAN architecture, the generator is linked to the loss function indirectly, via discriminator. The loss that we want to minimize is obtained by generator, encoder and the discriminator fed by generator and encoder. The generator is penalized for generating samples that discriminator component classifies as fake. The effect of the generator's parameters depends on the discriminator's parameters fed by the generator. Hence, the back-propagation begins from the output of the discriminator and proceeds through the discriminator to the generator. In this part of the training process, the discriminator is not updated. Generators and discriminators are highly dependent on each other. This dependency makes a hard problem for the training and even harder problem for the generator component.

Generators and discriminator are highly dependent on each other. In this paper, to reduce this dependency and force generator produce more reliable data that represents the real data, we proposed three different approaches. We used the BiGAN structure for anomaly detection proposed in (Zenati et al., 2018).

$$MSE = (\frac{1}{n})\sum_{i=1}^{n}(x_i - G(z_i))^2 \tag{3.6}$$

3.2.3.1 Method 1

We added extra training steps for generator independent from the discriminator, which updates only the generator's parameters by calculating the loss between the input of encoder and output of generator at the end of each epoch, illustrated in Algorithm 3. In the training phase, the generator aims to convince the discriminator so that generator can learn to generate more realistic data. To the end of each epoch, to increase the generator's learning performance, we added another custom loss function shown in equation 3.6, which updates only the generator's parameters. This loss is calculated, taking the mean squared error between the input of encoder (real data point) and the output of the generator.

3.2.3.2 Method 2

In addition to the proposed method in Algorithm 3, to improve the training and testing performance of BiGAN and to reduce the dependency of the generator, discriminator and encoder, we trained the generator component beforehand according to the equation 3.6. We feed the BiGAN with a pre-trained generator for simultaneous learning of generator, discriminator and encoder. Algorithm details are illustrated in Algorithm 4.

3.2.3.3 Method 3

In this approach, the generator is trained by minimizing the equation 3.6, before BiGAN training begins. Using a pre-trained generator in the BiGAN structure on an

Algorithm 3: Proposed BiGAN training structure

```
for number of training iterations do
    for k steps do
        Sample $z^t \sim p(z)$;
        Choose $x^t \in X$;
        $\hat{z} = E(x^t)$;
        $\hat{x} = G(z^t)$;
        Update $D(x, \hat{z})$ and $D(\hat{x}, z)$ by maximizing Eq. 3.2;
    end
    Sample $z^t \sim p(z)$;
    $\hat{x} = G(z^t)$;
    Update $G(z)$ and $E(\hat{x})$ by minimizing Eq. 3.2;
    Update $G(z)$ by minimizing Eq. 3.6;
end
```

anomaly detection task is proposed in this approach. We feed the BiGAN with a pre-trained generator for simultaneous learning of generator, discriminator and encoder. Algorithm details are illustrated in Algorithm 5.

Algorithm 4: Proposed BiGAN training structure with pre-trained and simultaneously updated generator

```
for number of training iterations do
    if the first training iteration then
        for l steps do
            Update $G(z)$ by minimizing Eq. 3.6;
        end
    end
    for k steps do
        Sample $z^t \sim p(z)$;
        Choose $x^t \in X$;
        $\hat{z} = E(x^t)$;
        $\hat{x} = G(z^t)$;
        Update $D(x, \hat{z})$ and $D(\hat{x}, z)$ by maximizing Eq. 3.2;
    end
    Sample $z^t \sim p(z)$;
    $\hat{x} = G(z^t)$;
    Update $G(z)$ and $E(\hat{x})$ by minimizing Eq. 3.2;
    Update $G(z)$ by minimizing Eq. 3.6;
end
```

Algorithm 5: Proposed BiGAN training structure with pre-trained generator

```
for number of training iterations do
    if the first training iteration then
        for $l$ steps do
            Update $G(z)$ by minimizing Eq. 3.6;
        end
    end
    for $k$ steps do
        Sample $z^t \sim p(z)$;
        Choose $x^t \in X$;
        $\hat{z} = E(x^t)$;
        $\hat{x} = G(z^t)$;
        Update $D(x, \hat{z})$ and $D(\hat{x}, z)$ by maximizing Eq. 3.2;
    end
    Sample $z^t \sim p(z)$;
    $\hat{x} = G(z^t)$;
    Update $G(z)$ and $E(\hat{x})$ by minimizing Eq. 3.2;
end
```

3.3 Model Evaluation Metrics

Since we need to overcome class imbalance problems, determining accuracy as an evaluation metric could mislead us about the results. The confusion matrix that defines precision and recall metrics is usually preferred for the imbalanced class problem.

One should take precision and recall into consideration together as an evaluation metric for the imbalanced class problem. We will compare implemented algorithms by their precision, recall, accuracy and F1 score metrics calculated from the confusion matrix. Precision means the percentage of our results, which are relevant. On the other hand, recall refers to the percentage of total relevant results correctly classified by our algorithm. We showed all metrics used in comparison in equation 3.7, 3.8, 3.9 and

3.10.

$$precision = \frac{TP}{TP + FP} \quad (3.7)$$

$$recall = \frac{TP}{TP + FN} \quad (3.8)$$

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (3.9)$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (3.10)$$

We tested all models on the same test set. We took One-Class SVM and BiGAN as our baseline methods. Since OC-SVM and BiGAN based approaches generate anomaly scores, they need a score threshold to classify the observations. In this study, we considered our test set as a validation set as well to determine the best threshold for each approach.

We classify the test set according to the percentile of generated scores in the test set. Any observation greater than the determined percentile of generated scores is labeled as an anomaly. We conduct exhaustive search and observed percentile zones to obtain the best F1 score while determining the threshold for each algorithm.

4 EXPERIMENT

4.1 Datasets and Descriptions

We evaluated our approaches and other state-of-the-art approaches on the KDDCUP99 10 percent dataset (Lichman, n.d.) and Credit Card Fraud Detection dataset (*Credit Card Fraud Detection - Anonymized credit card transactions labeled as fraudulent or genuine*, n.d.).

4.1.1 KDDCUP99 10 Percent Dataset

4.1.1.1 Overview

The KDDCUP99 data set is created for building a network intrusion detector, a predictive model capable of distinguishing between bad connections called intrusions or attacks, and good normal connections. It provides labeled observations according to attack type. Provided features' types and class distribution of the original data set are illustrated in the table 4.2 and 4.1 respectively.

4.1.1.2 Preprocessing

As a preprocessing step we applied one hot encoding to categorical features and normalized the data.

One Hot Encoding : Many machine learning algorithms require numerical values rather than categorical values. In the case that there is an ordinal relation between categories, turning categories into integer values may work. However, this approach affects the learning process in a negative way if there is no ordinal relation. In this case, one hot encoding is one of the most preferred approaches. According to one-hot encoding, categorical variables are represented as binary vectors. If the cardinality (the # of categories) of the categorical features is low (relative to the amount of data),

Table 4.1: Distribution of attacks and normal observations in data set.

Class	Ratio
smurf	0.568377
neptune	0.216997
normal	0.196911
back	0.004459
satan	0.003216
ipsweep	0.002524
portsweep	0.002105
warezclient	0.002065
teardrop	0.001982
pod	0.000534
nmap	0.000468
guess_passwd	0.000107
buffer_overflow	0.000061
land	0.000043
warezmaster	0.000040
imap	0.000024
rootkit	0.000020
loadmodule	0.000018
ftp_write	0.000016
multihop	0.000014
phf	0.000008
perl	0.000006
spy	0.000004

one-hot encoding gives reasonable results. We can use it as input into any model. In our data set, there are categorical and continuous features. To deal with categorical features in our data set, we implemented one-hot-encoding and enlarged the feature set.

Standardization : The result of standardization (or Z-score normalization), showed in equation 4.1, is that the features will be re-scaled so that they will have the properties of a standard normal distribution with zero mean and one variance. Standardizing the features so that they are centered around 0 with a standard deviation of 1 is not only important if we compare measurements with different units but also a general requirement for many machine learning algorithms. Intuitively, we can think of gradient descent as a prominent example (an optimization algorithm often used in logistic regression, SVMs, perceptrons, neural networks, etc.) with features being on different scales, and certain weights may update faster than others since the feature values x play a role in the weight updates. Since we use a neural network architecture, we conduct

Table 4.2: Features and types of KDDCUP99 10 Percent dataset

Column	Type
duration	continuous
protocol_type	symbolic
service	symbolic
flag	symbolic
src_bytes	continuous
dst_bytes	continuous
land	symbolic
wrong_fragment	continuous
urgent	continuous
hot	continuous
num_failed_logins	continuous
logged_in	symbolic
num_compromised	continuous
root_shell	continuous
su_attempted	continuous
num_root	continuous
num_file_creations	continuous
num_shells	continuous
num_access_files	continuous
num_outbound_cmds	continuous
is_host_login	symbolic
is_guest_login	symbolic
count	continuous
srv_count	continuous
serror_rate	continuous
srv_serror_rate	continuous
rerror_rate	continuous
srv_rerror_rate	continuous
same_srv_rate	continuous
diff_srv_rate	continuous
srv_diff_host_rate	continuous
dst_host_count	continuous
dst_host_srv_count	continuous
dst_host_same_srv_rate	continuous
dst_host_diff_srv_rate	continuous
dst_host_same_src_port_rate	continuous
dst_host_srv_diff_host_rate	continuous
dst_host_serror_rate	continuous
dst_host_srv_serror_rate	continuous
dst_host_rerror_rate	continuous
dst_host_srv_rerror_rate	continuous

feature-wise standardization by utilizing the formula in equation 4.1.

$$z = \frac{x - \mu}{\sigma} \tag{4.1}$$

4.1.1.3 Train and Test Set Preparation

Some attack types are rarely seen on data, and they come up with a class imbalance problem. We filtered mentioned attack types and aimed to predict correctly on the test set those attacks such as *pod*, *nmap*, *guesspasswd*, $bufferoverfl$, *land*, *warezmaster* and *normal*. In table 4.3, the distribution of data set labels is illustrated.

Since we train models with only normal observations, we include all anomalies in the test and validation set. We randomly select normal observation for the test and validation set. Eventually, 619 observations constitute our validation set containing 313 normal observations and 306 attack observations, and another 619 observations constitute our test set containing 306 normal observations and 313 attack observations. We use the validation set to determine the best score threshold since the experimented algorithms produce anomaly scores for each observation. In section 4.2, we elaborate on how we extract the best threshold from the validation set. Test, training, validation set do not include the same observations.

Table 4.3: Distribution of selected attacks and normal observations in data set.

Class	Ratio
normal	0.9937
pod	0.0027
nmap	0.0024
guess_passwd	0.0005
buffer_overflow	0.0003
land	0.0002
warezmaster	0.0002

4.1.2 Credit Card Fraud Detection Dataset

4.1.2.1 Overwiew

The credit card fraud dataset was downloaded from *Kaggle* (*Credit Card Fraud Detection - Anonymized credit card transactions labeled as fraudulent or genuine*, n.d.). It contains 492 fraudulent records out of 284807 transactions. Because of the privacy concerns, features were shared with the name of $V1, \ldots, V28$ as principle components acquired with PCA. Only time and amount features have not been transformed with PCA. Features types and class distribution are illustrated in table 4.4 and 4.5 respectively.

Table 4.5: Distribution of fraudulent and non-fraudulent observations in data set

Class	Ratio
non-fraudulent	0.998273
fraudulent	0.001727

4.1.2.2 Preprocessing

Since we did not tackle the problem as a time series problem, the time feature was not considered and dropped from the feature set. Due to the privacy concerns, features were shared with the name of $V1, \ldots, V28$ as principle components acquired with PCA. As a preprocessing step, we implement feature-wise standardization process by following equation 4.1.

4.1.2.3 Train and Test Set Preparation

Since we use a one-class algorithm, only non-fraudulent observations are included in the training set. We set class distribution of test set as illustrated in 4.6. The training set is constructed to include only non-fraudulent observations that are not in the test set.

Table 4.4: Features and types of Credit Card Fraud Detection dataset

Column	Type
Time	continuous
V1	continuous
V2	continuous
V3	continuous
V4	continuous
V5	continuous
V6	continuous
V7	continuous
V8	continuous
V9	continuous
V10	continuous
V11	continuous
V12	continuous
V13	continuous
V14	continuous
V15	continuous
V16	continuous
V17	continuous
V18	continuous
V19	continuous
V20	continuous
V21	continuous
V22	continuous
V23	continuous
V24	continuous
V25	continuous
V26	continuous
V27	continuous
V28	continuous
Amount	continuous

Since we train models with only normal observations, we include all anomalies in the test and validation set. We randomly select normal observation for the test and validation set. Eventually, 1241 observations constitute our validation set containing 990 normal observations and 241 fraudulent observations, and another 1242 observations constitute our test set containing 1001 normal observations and 241 fraudulent observations. We use the validation set to determine the best score threshold since the experimented algorithms produce anomaly scores for each observation. In section 4.2, we elaborate on how we extract the best threshold from the validation set. Test, training, validation set do not include the same observations.

Table 4.6: Distribution of fraudulent and non-fraudulent observations in test set.

Class	Ratio
non-fraudulent	0.805958
fraudulent	0.194042

4.2 Numerical Results

This section provides numerical results for the OC-SVM, BiGAN, and three proposed approaches introduced in subsections 3.2.3.

In this study, we discuss an anomaly detection problem. We converted the process of anomaly detection to a one-class classification task and trained algorithms using normal data only. Each algorithm discussed in this study generates an anomaly score for each observation and conducts the detection process by considering the anomaly scores. Since we trained all models on normal data, observations with higher anomaly scores are considered anomalies. After generating anomaly scores, to make classification, determining a threshold specified for each algorithm is needed.

We trained each model for 20 epochs. The classification task is conducted for each algorithm as follows. We used the validation set to determine the best anomaly score threshold. After generating a set of scores with each algorithm for the test and validation set, we determined the best score threshold as the anomaly score giving the best F1 score by conducting an exhaustive search. This process was conducted for each algorithm. We obtained different thresholds for each approach. We classified the observation according to the threshold of each algorithm. Observations greater than the corresponding threshold were labeled as an anomaly for the corresponding algorithm.

4.2.1 Results on KDDCUP99 10 Percent Dataset

619 observation constitutes our test set that contains 313 normal observations and 306 observations labeled as an attack. We classify the observation according to the threshold of each algorithm. Observations greater than the corresponding threshold are labeled as an anomaly for the corresponding algorithm. Table 4.7 shows the numerical results

and table 4.10 shows the run-time of each algorithm.

Our proposed three methods improved the performance of BiGAN. One-Class SVM gave the best precision score with low recall. When we pulled the recall of OC-SVM to %98 by changing the threshold, we noted that the precision of OC-SVM decreased dramatically to %65 implying that scores generated by OC-SVM are not as heterogeneous as the anomaly scores generated by BiGAN like approaches. Our first proposed method, algorithm 3, achieved an acceptable improvement when compared to BiGAN. Our second proposed method, algorithm 4, made drastic progress in the performance of BiGAN, giving the best F1 score, among other methods. Algorithm 4 and algorithm 5 gave similar results.

Table 4.7: Results according to the determined thresholds that render best F1 score on KDDCUP99 10 Percent dataset

Methods	Precision	Recall	Accuracy	F1-score
OC-SVM	**0.957**	0.752	0.853	0.842
BiGAN	0.7	0.915	0.764	0.793
BiGAN with Alg. 3	0.811	0.967	0.872	0.882
BiGAN with Alg. 4	0.836	**0.994**	**0.895**	**0.908**
BiGAN with Alg. 5	0.842	0.969	0.889	0.901

Table 4.8: Run-time in second of each algorithm experimented on KDDCUP99 10 Percent dataset.

Methods	Total run-time in second	1 epoch run-time for training BiGAN ($\mu \pm \sigma$)	1 epoch run-time for pre-trained generator ($\mu \pm \sigma$)
OC-SVM	738.58	-	-
BiGAN	76.66	3.79 ± 0.19	-
BiGAN with Alg. 3	90.24	5.24 ± 0.18	-
BiGAN with Alg. 4	87.71 + 90.32	5.24 ± 0.18	0.89 ± 0.06
BiGAN with Alg. 5	87.04 + 75.52	3.79 ± 0.19	0.89 ± 0.06

4.2.2 Results on Credit Card Fraud Detection Dataset

2483 observation constitutes our test set that contains 1991 normal observations and 492 observations labeled as fraudulent activities. We classify the observation according to the threshold of each algorithm. Observations greater than the corresponding threshold are labeled as an anomaly for the corresponding algorithm. Table 4.9 shows the numerical results.

Our proposed method algorithm 5, 4 and 3 improved performance of BiGAN. One-Class SVM gave the F1 score. Algorithm 3 which trains generator separately at the end of each epoch, 4 which uses pre-trained generator and trains generator separately at the end of each epoch, 5 which uses a pre-trained generator and trains all components only simultaneously, achieved an acceptable improvement when compared to BiGAN.

Table 4.9: Results according to the determined thresholds that render best F1 score on Credit Card Fraud Detection dataset

Methods	Precision	Recall	Accuracy	F1-score
OC-SVM	**0.859**	**0.868**	**0.946**	**0.863**
BiGAN	0.784	0.831	0.921	0.807
BiGAN with Algorithm 3	0.816	0.866	0.935	0.84
BiGAN with Algorithm 4	0.852	0.817	0.936	0.834
BiGAN with Algorithm 5	0.843	0.852	0.939	0.847

Table 4.10: Run-time in second of each algorithm experimented on Credit Card Fraud Detection dataset.

Methods	Total run-time in second	1 epoch run-time for training BiGAN ($\mu \pm \sigma$)	1 epoch run-time for pre-trained generator ($\mu \pm \sigma$)
OC-SVM	4816.72	-	-
BiGAN	209.13	10.47 ± 0.24	-
BiGAN with Alg. 3	251.34	13.26 ± 0.26	-
BiGAN with Alg. 4	231.06 + 250.47	13.26 ± 0.26	2.31 ± 0.46
BiGAN with Alg. 5	231.67 + 208.78	10.47 ± 0.24	2.31 ± 0.46

5 CONCLUSION AND FUTURE WORKS

In this study, we considered the network intrusion detection and credit card fraud detection tasks as anomaly detection tasks. Since the class imbalance problem is hard to overcome, we applied GAN based approaches, discussing the problem as a one-class problem.

Since the loss is calculated by the discriminator and the discriminator is fed by the output of generator and encoder, the effect of the generator's parameters depends on the impact of the discriminator's parameters. Hence, the back-propagation begins from the output of the discriminator and proceeds through the discriminator to the generator. The discriminator's parameter is not updated during this part of the training process. To reduce the effect of this dependency and to improve the generator's learning process and dependently discriminator's learning process, we applied three different training architecture for BiGAN and demonstrated that our three proposed approaches made progress on the performances of BiGAN.

According to our findings, we increased the performance of BiGAN on intrusion detection and credit card fraud detection tasks by changing the training architecture and adding extra training step for only updating the generator's parameters according to the loss described in the equation 3.6. Additionally, starting BiGAN training

with pre-trained generators contributed the performance on the test set more than the proposed algorithm 3 and 4 since we initialized the parameters of BiGAN's generator components by training generator on the training data beforehand. In this thesis, we demonstrated that the mentioned approaches increased the performance of BiGAN in the relatively low dimensional data set. We concluded that initializing the generators parameter by training it with the real data before BiGAN training and keeping updating the generator independent from discriminator at the end of each epoch had a major contribution to BiGAN training in anomaly detection tasks on KDDCUP99 10 Percent and Credit Card Fraud Detection data set. Starting the BiGAN training with pre-trained generator as in algorithm 5 contributed to the performance of BiGAN the most on both data set. We conclude from our experiments that BiGAN for anomaly detection setup is affected positively in terms of performance on test set when it is trained with our proposed approaches.

In this study, we added the mean squared error term to the generator's training process. In algorithm 3, we updated the generator at the end of each epoch, in algorithm 4 we pre-trained and updated generator and in algorithm 5 we only pre-trained the generator. On all of these three methods, we used the mean squared error. In N dimensional signal space, a signal is represented by signal points. If any distortion vector is added to a signal point with the same length will result in an equal mean squared error. In this thesis, we studied two different, relatively low dimensional data sets. We left an open point to analyze how proposed approaches result on the high dimensional feature sets. Apart from the mean squared error, different loss metrics can be applied to deal with drawbacks of mean squared error and make progress on the BiGAN performance on anomaly detection tasks.

There are different loss terms introduced for increasing the performance of GANs. They are mainly conducted on computer vision tasks. Applying approaches presented in this study on the BiGAN version of corresponding GANs architectures such as Wasserstein GAN (Arjovsky et al., 2017a), least-squares GAN (Mao et al., 2017), loss-sensitive GAN (Qi, 2019) is another future research direction.

Various anomaly detection tasks require multi-class classification. In this study, binary classification is conducted by labeling all different attack types as an anomaly observation. Our approach only considers all types of attacks as an anomaly and does not classify them according to their type. In future works, these methods can be studied to handle multi-class anomaly detection tasks.

REFERENCES

Agrawal, R., Gehrke, J., Gunopulos, D. and Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications, *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pp. 94–105.

Akcay, S., Atapour-Abarghouei, A. and Breckon, T. P. (2018). Ganomaly : Semi-supervised anomaly detection via adversarial training, *Asian Conference on Computer Vision*, Springer, pp. 622–637.

An, J. and Cho, S. (2015). Variational autoencoder based anomaly detection using reconstruction probability, *Special Lecture on IE* **2**(1).

Angiulli, F. and Pizzuti, C. (2002). Fast outlier detection in high dimensional spaces, *European Conference on Principles of Data Mining and Knowledge Discovery*, Springer, pp. 15–27.

Arjovsky, M., Chintala, S. and Bottou, L. (2017a). Wasserstein gan, *arXiv preprint arXiv :1701.07875* .

Arjovsky, M., Chintala, S. and Bottou, L. (2017b). Wasserstein gan, *arXiv preprint arXiv :1701.07875* .

Baur, C., Wiestler, B., Albarqouni, S. and Navab, N. (2018). Deep autoencoding models for unsupervised anomaly segmentation in brain mr images, *International MICCAI Brainlesion Workshop*, Springer, pp. 161–169.

Bian, J., Hui, X., Sun, S., Zhao, X. and Tan, M. (2019). A novel and efficient cvae-gan-based approach with informative manifold for semi-supervised anomaly detection, *IEEE Access* **7** : 88903–88916.

Breunig, M. M., Kriegel, H.-P., Ng, R. T. and Sander, J. (2000). Lof : identifying density-based local outliers, *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104.

Brito, M., Chavez, E., Quiroz, A. and Yukich, J. (1997). Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection, *Statistics & Probability Letters* **35**(1) : 33–42.

Chandola, V., Banerjee, A. and Kumar, V. (2009). Anomaly detection : A survey, *ACM computing surveys (CSUR)* **41**(3) : 1–58.

Che, T., Li, Y., Jacob, A. P., Bengio, Y. and Li, W. (2016). Mode regularized generative adversarial networks, *arXiv preprint arXiv :1612.02136* .

Chen, F., Liu, Z. and Sun, M.-t. (2015). Anomaly detection by using random projection forest, *2015 IEEE International Conference on Image Processing (ICIP)*, IEEE, pp. 1210–1214.

Chen, J., Shen, Y. and Ali, R. (2018). Credit card fraud detection using sparse autoencoder and generative adversarial network, *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, IEEE, pp. 1054–1059.

Chen, X. and Konukoglu, E. (2018). Unsupervised detection of lesions in brain mri using constrained adversarial auto-encoders, *arXiv preprint arXiv :1806.04972* .

Credit Card Fraud Detection - Anonymized credit card transactions labeled as fraudulent or genuine (n.d.). `https://www.kaggle.com/mlg-ulb/creditcardfraud`. Accessed : 2019-09-30.

Donahue, J., Krähenbühl, P. and Darrell, T. (2016). Adversarial feature learning, *arXiv preprint arXiv :1605.09782* .

Dong, H., Neekhara, P., Wu, C. and Guo, Y. (2017). Unsupervised image-to-image translation with generative adversarial networks, *arXiv preprint arXiv :1701.02676* .

Emmott, A. F., Das, S., Dietterich, T., Fern, A. and Wong, W.-K. (2013). Systematic construction of anomaly detection benchmarks from real data, *Proceedings of the ACM SIGKDD workshop on outlier detection and description*, pp. 16–21.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X. et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise., *Kdd*, Vol. 96, pp. 226–231.

Fiore, U., De Santis, A., Perla, F., Zanetti, P. and Palmieri, F. (2019). Using generative adversarial networks for improving classification effectiveness in credit card fraud detection, *Information Sciences* **479** : 448–455.

Gan, Z., Chen, L., Wang, W., Pu, Y., Zhang, Y., Liu, H., Li, C. and Carin, L. (2017). Triangle generative adversarial networks, *Advances in neural information processing systems*, pp. 5247–5256.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014). Generative adversarial nets, *Advances in neural information processing systems*, pp. 2672–2680.

Guha, S., Rastogi, R. and Shim, K. (1998). Cure : an efficient clustering algorithm for large databases, *ACM Sigmod record* **27**(2) : 73–84.

Hautamaki, V., Karkkainen, I. and Franti, P. (2004). Outlier detection using k-nearest neighbour graph, *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, Vol. 3, IEEE, pp. 430–433.

He, Z., Xu, X. and Deng, S. (2003). Discovering cluster-based local outliers, *Pattern Recognition Letters* **24**(9-10) : 1641–1650.

Hinneburg, A., Keim, D. A. et al. (1998). An efficient approach to clustering in large multimedia databases with noise, *KDD*, Vol. 98, pp. 58–65.

Jaiswal, A., AbdAlmageed, W., Wu, Y. and Natarajan, P. (2018). Bidirectional conditional generative adversarial networks, *Asian Conference on Computer Vision*, Springer, pp. 216–232.

Jin, W., Tung, A. K., Han, J. and Wang, W. (2006). Ranking outliers using symmetric neighborhood relationship, *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, pp. 577–593.

Kim, T., Cha, M., Kim, H., Lee, J. K. and Kim, J. (2017). Learning to discover cross-domain relations with generative adversarial networks, *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, pp. 1857–1865.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes, *arXiv preprint arXiv :1312.6114* .

Knox, E. M. and Ng, R. T. (1998). Algorithms for mining distancebased outliers in large datasets, *Proceedings of the international conference on very large data bases*, Citeseer, pp. 392–403.

Kriegel, H.-P., Kröger, P., Schubert, E. and Zimek, A. (2009). Loop : local outlier probabilities, *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 1649–1652.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z. et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4681–4690.

Li, C., Liu, H., Chen, C., Pu, Y., Chen, L., Henao, R. and Carin, L. (2017). Alice : Towards understanding adversarial learning for joint distribution matching, *Advances in Neural Information Processing Systems*, pp. 5495–5503.

Lichman, M. (n.d.). Uci machine learning repository, `http://archive.ics.uci.edu/ml`. Accessed : 2019-09-30.

Liu, F. T., Ting, K. M. and Zhou, Z.-H. (2008). Isolation forest, *2008 Eighth IEEE International Conference on Data Mining*, IEEE, pp. 413–422.

Luc, P., Couprie, C., Chintala, S. and Verbeek, J. (2016). Semantic segmentation using adversarial networks, *arXiv preprint arXiv :1611.08408* .

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations, *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1, Oakland, CA, USA, pp. 281–297.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I. and Frey, B. (2015). Adversarial autoencoders, *arXiv preprint arXiv :1511.05644* .

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z. and Paul Smolley, S. (2017). Least squares generative adversarial networks, *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2794–2802.

Mescheder, L., Nowozin, S. and Geiger, A. (2017). Adversarial variational bayes : Unifying variational autoencoders and generative adversarial networks, *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, pp. 2391–2400.

Metz, L., Poole, B., Pfau, D. and Sohl-Dickstein, J. (2016). Unrolled generative adversarial networks, *arXiv preprint arXiv :1611.02163* .

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets, *arXiv preprint arXiv :1411.1784* .

Ng, R. T. and Han, J. (n.d.). Efficient and effective clustering methods for spatial data mining.

Papadimitriou, S., Kitagawa, H., Gibbons, P. B. and Faloutsos, C. (2003). Loci : Fast outlier detection using the local correlation integral, *Proceedings 19th international conference on data engineering (Cat. No. 03CH37405)*, IEEE, pp. 315–326.

Parzen, E. (1962). On estimation of a probability density function and mode, *The annals of mathematical statistics* **33**(3) : 1065–1076.

Perarnau, G., Van De Weijer, J., Raducanu, B. and Álvarez, J. M. (2016). Invertible conditional gans for image editing, *arXiv preprint arXiv :1611.06355* .

Qi, G.-J. (2019). Loss-sensitive generative adversarial networks on lipschitz densities, *International Journal of Computer Vision* pp. 1–23.

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

Ramaswamy, S., Rastogi, R. and Shim, K. (2000). Efficient algorithms for mining outliers from large data sets, *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 427–438.

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B. and Lee, H. (2016). Generative adversarial text to image synthesis, *arXiv preprint arXiv :1605.05396* .

Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E. and Kloft, M. (2018). Deep one-class classification, *International conference on machine learning*, pp. 4393–4402.

Sabokrou, M., Fathy, M. and Hoseini, M. (2016). Video anomaly detection and localisation based on the sparsity and reconstruction error of auto-encoder, *Electronics Letters* **52**(13) : 1122–1124.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A. and Chen, X. (2016). Improved techniques for training gans, *Advances in neural information processing systems*, pp. 2234–2242.

Schlegl, T., Seeböck, P., Waldstein, S. M., Langs, G. and Schmidt-Erfurth, U. (2019). f-anogan : Fast unsupervised anomaly detection with generative adversarial networks, *Medical image analysis* **54** : 30–44.

Schlegl, T., Seeböck, P., Waldstein, S. M., Schmidt-Erfurth, U. and Langs, G. (2017). Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, *International Conference on Information Processing in Medical Imaging*, Springer, pp. 146–157.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J. and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution, *Neural computation* **13**(7) : 1443–1471.

Schölkopf, B., Platt, J. C. and Smola, A. J. (2000). Kernel method for percentile feature extraction.

Sheikholeslami, G., Chatterjee, S. and Zhang, A. (1998). Wavecluster : A multi-resolution clustering approach for very large spatial databases, *VLDB*, Vol. 98, pp. 428–439.

Siddiqui, M. A., Fern, A., Dietterich, T. G. and Das, S. (2016). Finite sample complexity of rare pattern anomaly detection., *UAI*.

Tan, S. C., Ting, K. M. and Liu, T. F. (2011). Fast anomaly detection for streaming data, *Twenty-Second International Joint Conference on Artificial Intelligence*.

Tang, J., Chen, Z., Fu, A. W.-C. and Cheung, D. W. (2002). Enhancing effectiveness of outlier detections for low density patterns, *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, pp. 535–548.

Tax, D. M. and Duin, R. P. (2004). Support vector data description, *Machine learning* **54**(1) : 45–66.

Vapnik, V. (1963). Pattern recognition using generalized portrait method, *Automation and remote control* **24** : 774–780.

Vapnik, V. and Chervonenkis, A. (1974). Theory of pattern recognition.

Wang, W., Yang, J., Muntz, R. et al. (1997). Sting : A statistical information grid approach to spatial data mining, *VLDB*, Vol. 97, pp. 186–195.

Warde-Farley, D. and Bengio, Y. (2016). Improving generative adversarial networks with denoising feature matching.

Wu, K., Zhang, K., Fan, W., Edwards, A. and Philip, S. Y. (2014). Rs-forest : A rapid density estimator for streaming anomaly detection, *2014 IEEE International Conference on Data Mining*, IEEE, pp. 600–609.

Yeh, R. A., Chen, C., Yian Lim, T., Schwing, A. G., Hasegawa-Johnson, M. and Do, M. N. (2017). Semantic image inpainting with deep generative models, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5485–5493.

Yi, Z., Zhang, H., Tan, P. and Gong, M. (2017). Dualgan : Unsupervised dual learning for image-to-image translation, *Proceedings of the IEEE international conference on computer vision*, pp. 2849–2857.

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

Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X. and Metaxas, D. N. (2017). Stackgan : Text to photo-realistic image synthesis with stacked generative adversarial networks, *Proceedings of the IEEE international conference on computer vision*, pp. 5907–5915.

Zhang, T., Ramakrishnan, R. and Livny, M. (1996). Birch : an efficient data clustering method for very large databases, *ACM Sigmod Record* **25**(2) : 103–114.

Zhao, J., Mathieu, M. and LeCun, Y. (2016). Energy-based generative adversarial network, *arXiv preprint arXiv :1609.03126* .

Zhou, C. and Paffenroth, R. C. (2017). Anomaly detection with robust deep autoencoders, *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp. 665–674.

Zhu, J.-Y., Park, T., Isola, P. and Efros, A. A. (2017a). Unpaired image-to-image translation using cycle-consistent adversarial networks, *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232.

Zhu, J.-Y., Park, T., Isola, P. and Efros, A. A. (2017b). Unpaired image-to-image translation using cycle-consistent adversarial networks, *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232.

APPENDIX A MODEL ARCHITECTURE OF KDDCUP99 10 PERCENT DATA SET

APPENDIX A.1 Generator

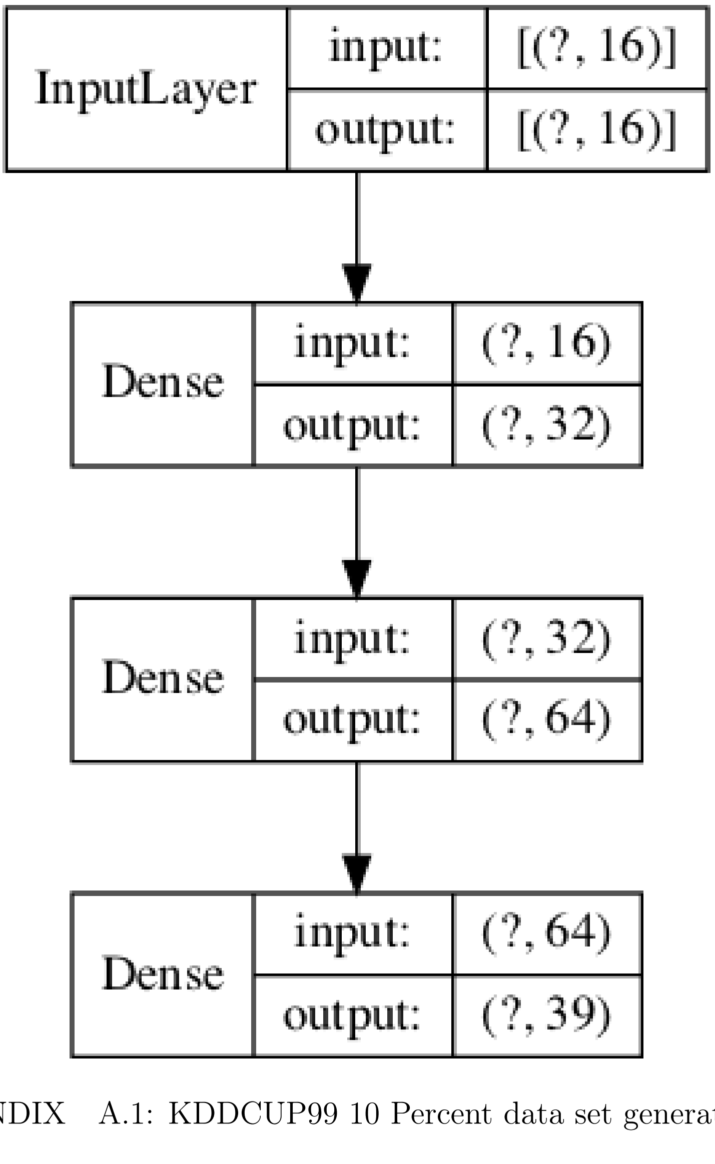


Figure APPENDIX A.1: KDDCUP99 10 Percent data set generator architecture

APPENDIX A.2 Encoder

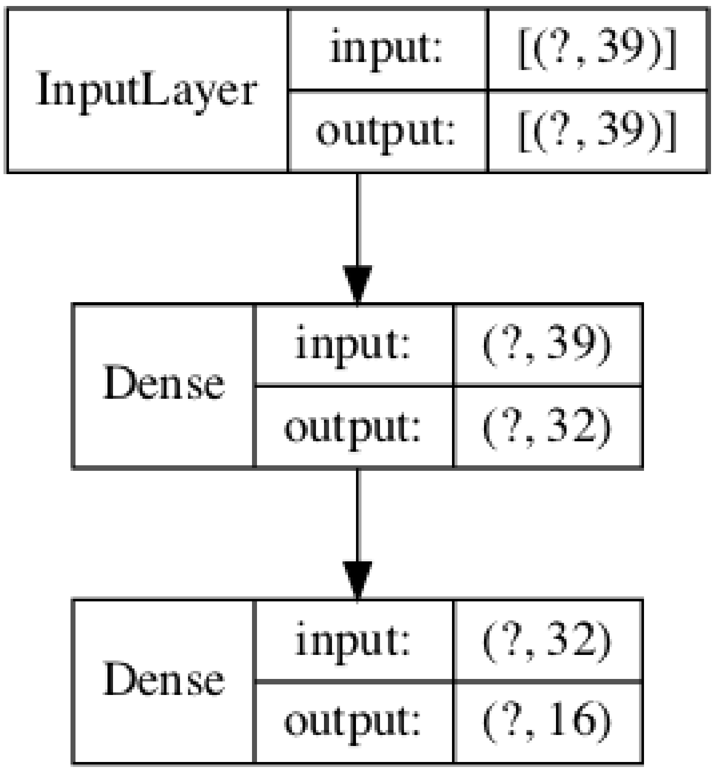


Figure APPENDIX A.2: KDDCUP99 10 Percent data set encoder architecture

APPENDIX A.3 Discriminator

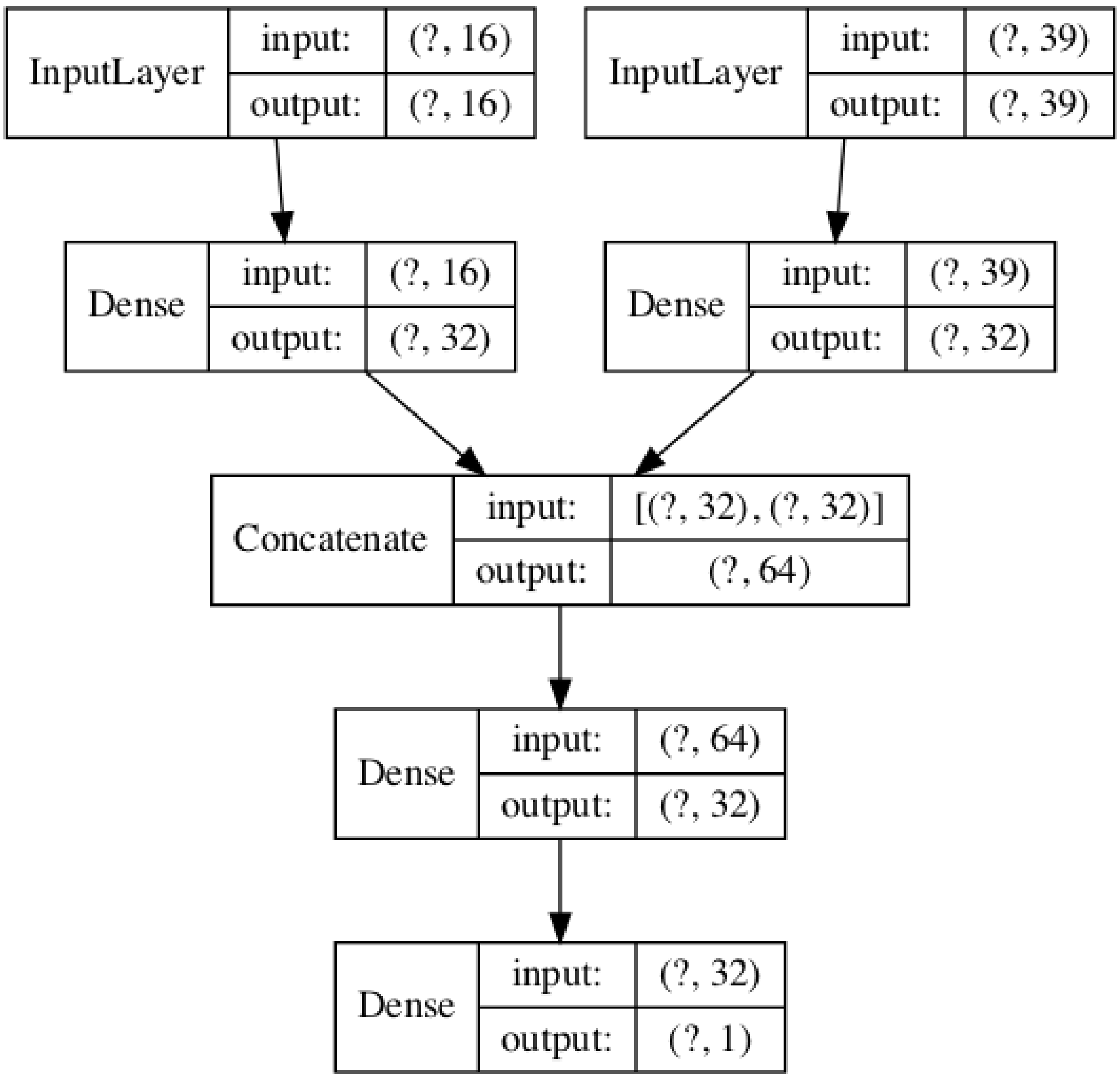


Figure APPENDIX A.3: KDDCUP99 10 Percent data set discriminator architecture

APPENDIX B MODEL ARCHITECTURE OF CREDIT CARD FRAUD DETECTION DATA SET

APPENDIX B.1 Generator

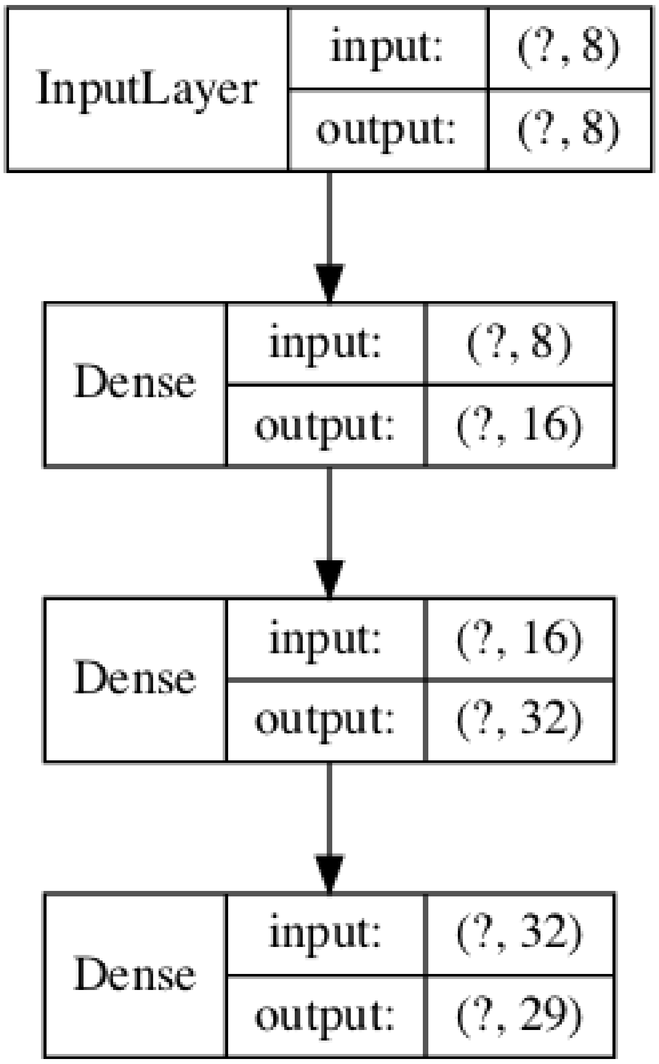


Figure APPENDIX B.1: Credit Card Fraud Detection data set generator architecture

APPENDIX B.2 Encoder

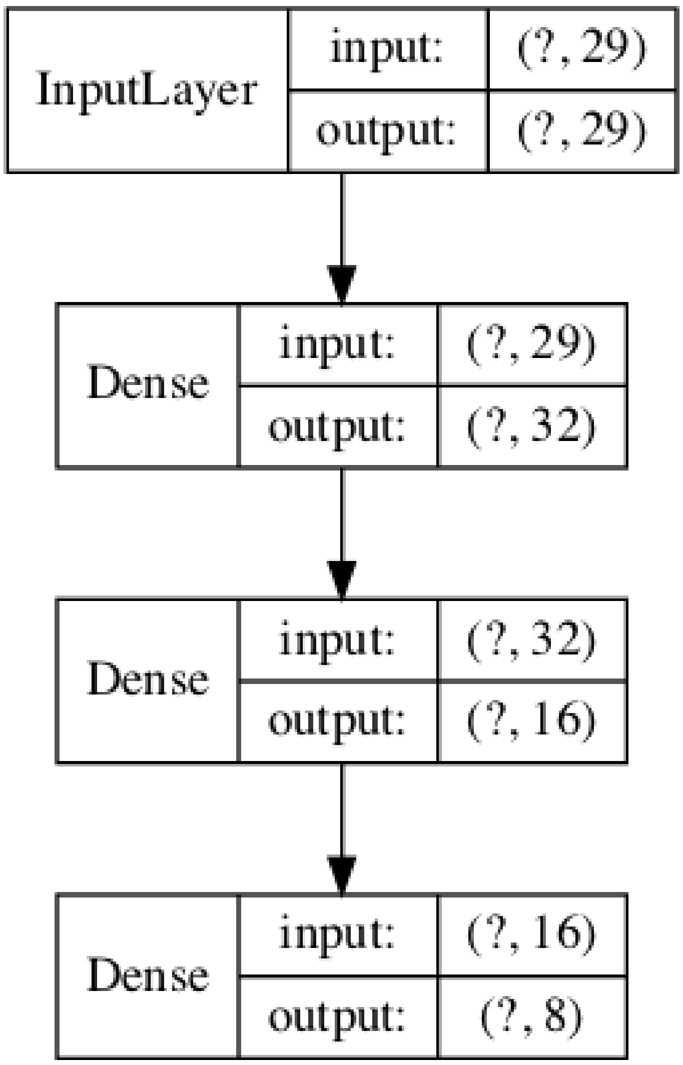


Figure APPENDIX B.2: Credit Card Fraud Detection data set encoder architecture

APPENDIX B.3 Discriminator

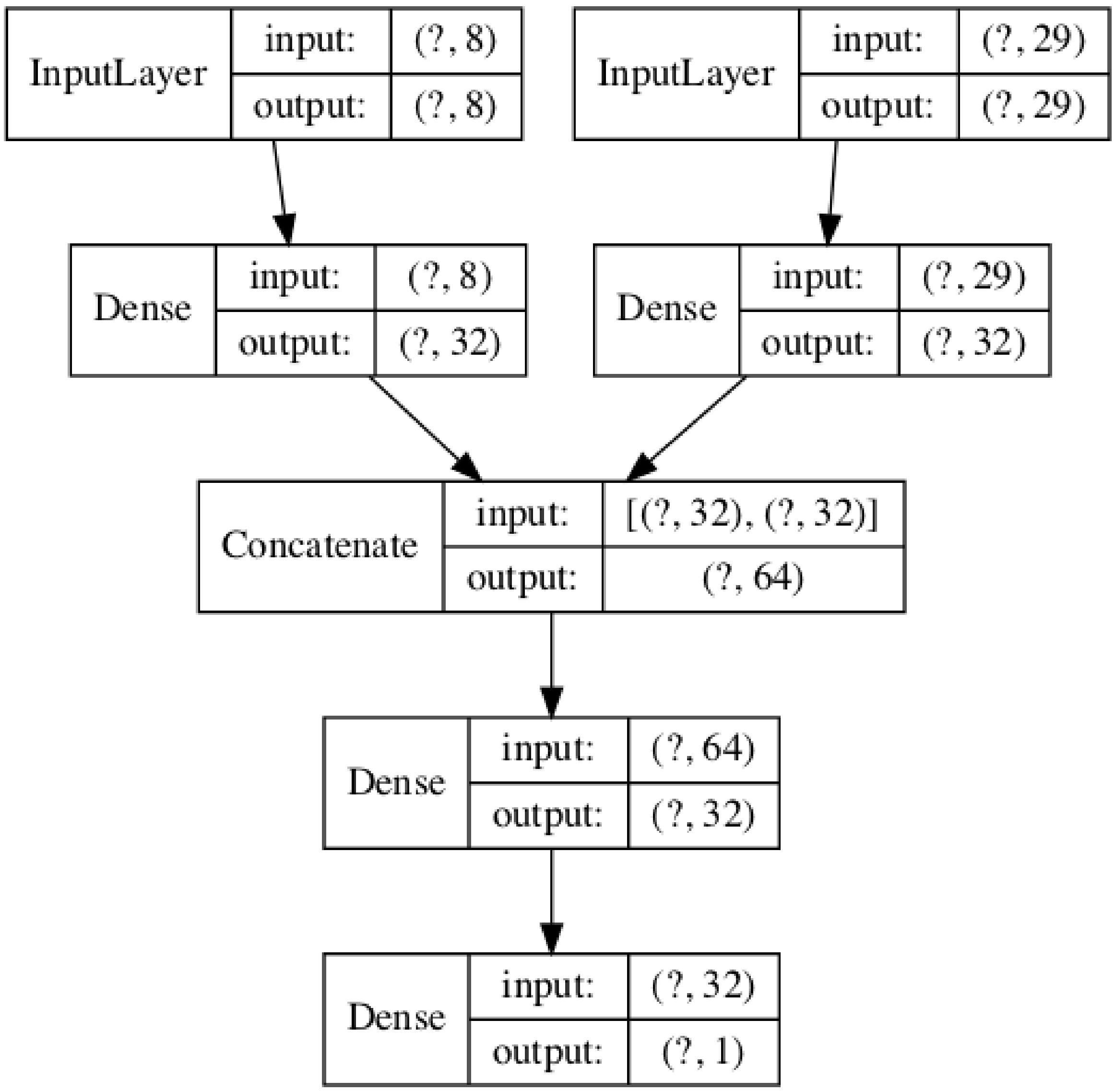


Figure APPENDIX B.3: Credit Card Fraud Detection data set discriminator architecture

BIOGRAPHICAL SKETCH

Muhammet Oğuz Kaplan was born in Istanbul on February 17, 1994. He studied at Turkish Naval High School where he was graduated in 2012. He attended the undergraduate program of Industrial Engineering at Istanbul Technical University. He received his B.S. degree in the Industrial Engineering in 2016. He worked at B/S/H and SIEMENS in the early years of the professional career. Now, he works at Tarfin as a Data Scientist since May 2019 and continues to work towards a master's degree in Industrial Engineering under the supervision of Assoc. Prof. Dr. Sadettin Emre Alptekin the Institute of Science and Engineering, Galatasaray University.

PUBLICATIONS

— Kaplan, M. O., Alptekin, S. E. (Accepted). An improved BiGAN based approach for anomaly detection. In 2020 24th International Conference on Knowledge-Based and Intelligent Information Engineering Systems (KES).