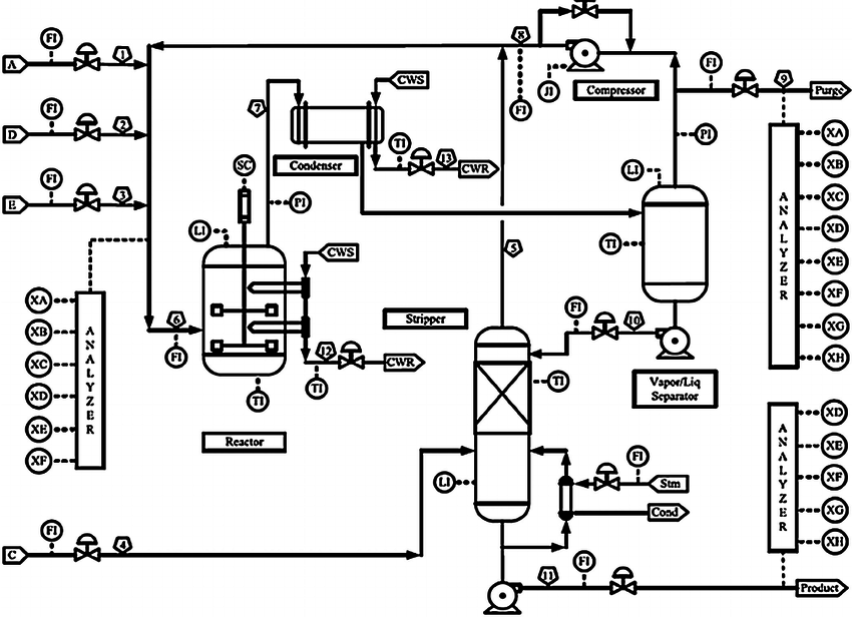
# ~~Introduction~~

## ~~Context~~

~~The machine learning (ML) area has undergone radical change in the last several years. Academia and industry have invested billions of dollars in developing massively complex machine learning algorithms. Deep learning (DL), which deploys multiple layers in the network, achieved great success in image classification (~~*~~Ref. ImageNet Classification with Deep Convolutional Neural Networks~~*~~), natural language processing (~~*~~Ref. Very Deep Transformers for Neural Machine Translation~~*~~), etc. One fundamental difference between deep learning and traditional machine learning algorithms is the scalability of data volume. In other words, deep learning models can continuously improve performance given more data and become a prioritized method for big data processing. One application scenario of big data processing is anomaly detection in industrial production.~~

~~The industrial processes have gradually become extremely complex to fulfill the requirement of high production efficiency and quality. More and more monitor and controllable variables are introduced, which pose challenges for system control. Early detection of anomalies and implementation of correspondence measures are critical for safety and effective operation. For anomaly detection, as discussed in (~~*~~Ref. A survey of the application of basic data-driven and model-based methods in process monitoring and fault diagnosis~~*~~) discussed, the methods are generally classified into two types: data-driven and model-based. The latter usually requires expert-level knowledge and experience to build the sophisticated model, which is difficult to achieve in reality. Compared to the model-based method, the data-driven method such as machine learning relies on analyzing large amounts of data, which provides a practical solution for various industrial productions. Due to the feasibility and practicability, machine learning has become an essential analysis method in the industrial process.~~

~~This thesis investigates machine learning methods for anomaly detection in Tennessee Eastman Process (TEP). The TEP proposed in (~~*~~Ref. A PLANT-WIDE INDUSTRIAL PROCESS PROBLEM~~*~~) is an industrial chemical process model and can be used as a benchmark model to evaluate the control system's effectiveness. The TEP mainly consists of five process units: a reactor where an exothermic reaction occurs, a product condenser, a vapor-liquid separator, a compressor and a stripper. The process diagram is shown in (~~*~~fig~~*~~). There are also simplified TEP versions as (~~*~~Ref. Model predictive control of a continuous, nonlinear, two-phase reactor~~*~~) proposed.~~

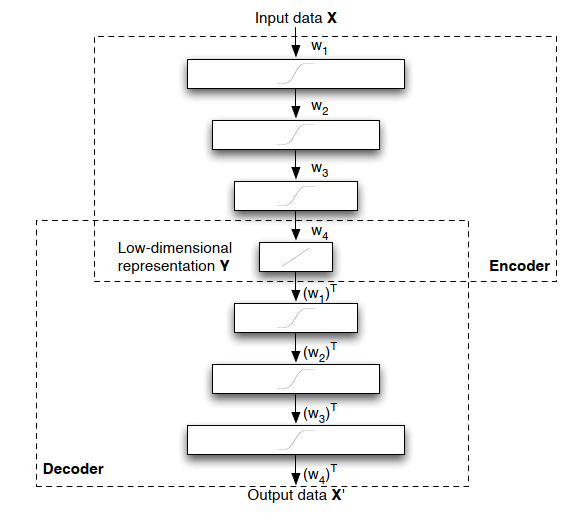
~~~~

~~(~~*~~Ref. A PLANT-WIDE INDUSTRIAL PROCESS PROBLEM~~*~~)~~

~~In general, TEP is a nonlinear unstable system with multi-input/outputs variables and characterized by fast and slow dynamic behaviors (~~*~~Ref. Process in Modelica Process in Modelica Process in Modelica~~*~~). Due to this reason, the data generated in TEP has the following characteristics:~~

* ~~high dimensionality,~~
* ~~nonlinearity,~~
* ~~non-Gaussian distribution.~~

~~High-dimensional data pose a severe challenge for machine learning algorithms, and some algorithms may even break down in practice. In (~~*~~Ref. A Few Useful Things to Know about Machine Learning~~*~~), they gave two descriptive explanations for the~~ *~~curse of dimensionality~~*~~. The first explanation is, the influence of noise accumulates as dimensionality increases, i.e., the noise from irrelated features may suppress relevant features' influence. The second explanation is that the similarity between data points increases as dimension expands. For example, when using distance as criteria to decide if data points are neighbors, the number of neighbors of data points increase as the dimensionality increases. To mitigate the impact of high dimensionality, feature selection~~ ~~(~~*~~Ref. Feature Engineering and Selection~~*~~) and feature extraction (dimension reduction) are proposed (~~*~~Ref. Fault Detection of the Tennessee Eastman Process Using Improved PCA and Neural Classifier~~*~~). The concept of feature selection is to select the most relevant features based on correlation metrics and expect to improve the machine learning model's performance (~~*~~Ref. Feature Selection for Fault Detection Systems: Application to the Tennessee Eastman Process~~*~~). The feature extraction represents the high dimensional data into the lower dimension and extracts the hidden structures (~~*~~Ref. A Tutorial on Principal Component Analysis~~*~~). As a nonlinear feature extraction method, autoencoder (AE) use the encoder part to represent the input data into a low dimension in the middle hidden layers (~~*~~bottleneck),~~* ~~then uses the decoder part to map the data back in high dimensional space (~~*~~Ref. Dimensionality Reduction: A Comparative Review~~*~~), as (~~*~~fig~~*~~) shows. For autoencoders, it is trained to copy the input to its output. One key point of this thesis is to explore the application of autoencoder in TEP's anomaly detection.~~

~~~~

~~(~~*~~Ref. Dimensionality Reduction: A Comparative Review~~*~~)~~

~~??? To be added~~

## ~~Problem Statement~~

~~The main goal of this research is to evaluate the feasibility of deep learning, especially the autoencoder, in TEP's anomaly detection. Analyzing the performance of different machine learning models can help enhance the understanding of anomaly detection and provide the possibility for further improvement. As the literature research implies, there is no detailed work comparing different anomaly detection methods for TEP. This thesis seeks to fill the gaps.~~

## ~~Research Questions~~

~~This thesis's main contribution is to provide a comprehensive comparison between the traditional machine learning method and the deep learning method for TEP anomaly detection. Then analyze the reasons for the difference and implement an improved deep learning model for anomaly detection. More specifically, this research is to answer the following questions:~~

* ~~What is the current status of research in TEP's anomaly detection, and what machine learning algorithms are already applied?~~
* ~~Develop evaluation metrics and compare the performance of different algorithms using the TEP dataset.~~
* ~~Analyze the reasons for the performance difference. Based on the evaluation results, implement a deep learning method.~~

## ~~Organization (may change)~~

~~Chapter 1 introduces the topic and provides an overview of the problem. Chapter 2 reviews the background and prior works about machine learning algorithms. Chapter 3 describes the structure and function of machine learning models for the experiment in detail. Chapter 4 includes the setting of the experiment, the adjustment of model parameters and result analysis. Chapter 5 summarizes the works and points out the future research directions.~~

# ~~Background Theory~~

~~This chapter provides the necessary background and theoretical knowledge to understand the methodology in chapter 4. It first introduces the basic concepts of machine learning, then gives a broad overview of traditional machine learning and deep learning. Finally, it reviews the prior work and compares the performance of different machine learning methods.~~

## ~~Machine Learning Concept~~

~~It is not a trivial task to give a precise definition of machine learning. If a machine can realize self-adjustment and improve the performance given more external information, we might say it learns (~~*~~Ref. INTRODUCTION TO MACHINE LEARNING - Nils J. Nilsson~~*~~). Machine learning is one of the fast-evolving areas in both academia and industry in the last decades. As an important branch of machine learning, deep learning has achieved huge success in pattern recognition and translation (~~*~~Ref. The Unreasonable Effectiveness of Deep Learning in Artificial Intelligence~~*~~). This is due to several reasons: 1) deep learning methods can realize sustained performance improvement when given more data exceeding the threshold of older learning algorithms (~~*~~fig below~~*~~); 2) deep learning can reconstruct the unknown structure and is based on the distribution of inputs to find appropriate representation (~~*~~Ref. Deep Learning of Representations for Unsupervised and Transfer Learning~~*~~).~~

~~~~

~~(~~*~~Ref.~~* [~~https://www.slideshare.net/ExtractConf~~](https://www.slideshare.net/ExtractConf)~~)~~

~~Machine learning algorithms can be classified into different groups based on the criteria. By learning style (~~*~~Ref. A Survey on Machine Learning: Concept, Algorithms and Applications~~*~~), it could be categorized into:~~

* ~~Supervised learning: Each input data has a corresponded label, e.g., Positive/Negative. The machine learning algorithm (classifier) can find the mapping between input data and labels by training. Further, the trained classifier can predict the label of data.~~
* ~~Unsupervised learning: The input data has no label, unlike in supervised learning. Machine learning algorithms' task is to find the typical input data pattern and cluster it into groups.~~
* ~~Semi-supervised learning: This can be seen as a hybrid version of supervised learning and unsupervised learning because the input data is only partially labeled. The task is to predict the label of the data and execute the clustering.~~
* ~~Reinforcement learning: The algorithms are trained to choose behavior in a specific environment to obtain the most reward.~~

~~The machine learning algorithms for TEP's anomaly detection can also be classified into traditional- and deep learning algorithms in view of prior work.~~

* ~~Traditional algorithms: Principal Component Analysis (PCA) (~~*~~Ref. Fault Detection of the Tennessee Eastman Process Using Improved PCA and Neural Classifier~~*~~), Support Vector Machine (SVM) (~~*~~Ref. Study on Support Vector Machine-Based Fault Detection in Tennessee Eastman Process~~*~~), Clustering (~~*~~Ref. Unsupervised Feature Selection Based on Fuzzy Clustering for Fault Detection of the Tennessee Eastman Process~~*~~), Random Forest Tree (~~*~~Ref. A Dynamic Nonlinear Process Fault Diagnosis Method Using Canonical Rotation Forest~~*~~), ???adding more~~
* ~~Deep learning algorithms:~~ ~~Long Short-Term Memory Recurrent Neural Network(~~*~~Ref. Fault Detection and Diagnosis in a Chemical Process using Long Short-Term Memory Recurrent Neural Network~~*~~),~~ ~~Stacked Sparse Autoencoders (~~*~~Ref. Mutual Information−Dynamic Stacked Sparse Autoencoders for Fault Detection~~*~~), Variational Autoencoders (~~*~~Ref. Comparison of Semi-supervised Deep Neural Networks for Anomaly Detection in Industrial Processes~~*~~), ???adding more~~

~~Certain traditional algorithms are primarily based on statistical knowledge or assumption and have some limitations. For example, PCA assumes that the process data are linear correlated and in Gaussian distribution (~~*~~Ref. A Tutorial on Principal Component Analysis~~*~~). Another limitation of PCA is the information loss during the projection of high-dimensional data into low-dimensional data (~~*~~Ref. A NONLINEAR SUPPORT VECTOR MACHINE BASED FEATURE SELECTION APPROACH FOR FAULT DETECTION AND DIAGNOSIS: APPLICATION TO THE TENNESSEE EASTMAN PROCESS~~*~~). SVM solves the nonlinear and non-Gaussian distribution issues by transforming the data in a high dimension space with a kernel function. Then SVM uses the optimal hyperplane to separate the data and makes the classification (~~*~~Ref. A NONLINEAR SUPPORT VECTOR MACHINE BASED FEATURE SELECTION APPROACH FOR FAULT DETECTION AND DIAGNOSIS: APPLICATION TO THE TENNESSEE EASTMAN PROCESS~~*~~). But the SVM classifier's performance depends strongly on the kernel function and is prone to overfitting (~~*~~Ref. Overcome Support Vector Machine Diagnosis Overfitting~~*~~). Compared to traditional algorithms, deep learning algorithms composed of the artificial neural network have many advantages and provide a universal solution for anomaly detection (~~*~~Ref. Approximation with Artificial Neural Networks~~*~~).~~

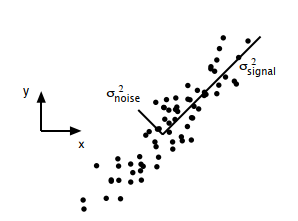
## ~~Traditional algorithms~~

~~The traditional machine learning algorithms in this thesis refers to the algorithms in which no artificial neural network is involved. The purpose of the distinction is to investigate the impact of structural complexity, i.e., if adding more layers in deep learning can improve performance in anomaly detection.~~

### ~~Principal component analysis~~

~~As a feature extraction (dimension reduction) method, the principal component analysis's goal is to represent the data in such a way that only the most~~ *~~meaningful~~* ~~features remain. More specifically, the representation of data is achieved by projecting the data on a new orthonormal basis. The assumption for the projecting of data is the linearity of data, i.e., the data are the linear combination of the basis vectors. For example, we want to transform the 2-dimensional data points into 1-dimensional data points. Signal-to-noise ratio (SNR) is used as a measure to evaluate the meaningfulness of data points in an axis direction.~~

~~The axis is defined as the signal (meaningful) direction because the data points have the largest variances in that direction. The axis is defined as the noise (meaningless) direction in which the data points have the smallest variance. Those two axes are orthogonal. As the (~~*~~figure~~*~~) shows, a meaningful way of representing the data is to project the data in axis direction and ignore the projection of data in axis. In this way, we get the 1-dimensional representation of the data points.~~

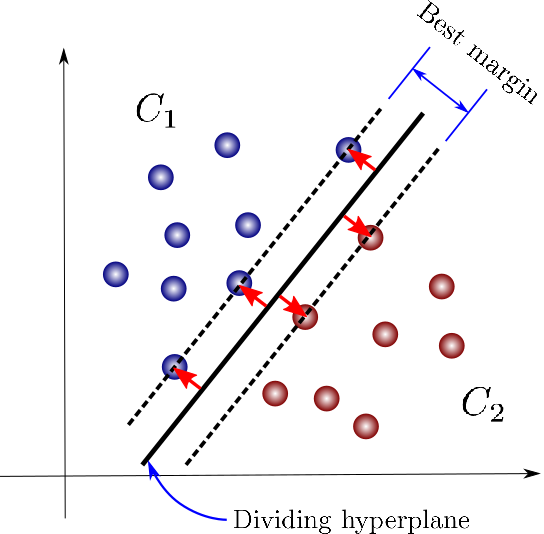
~~~~

~~(~~*~~Ref. A Tutorial on Principal Component Analysis~~*~~)~~

~~Although PCA is a non-parametric approach and easy to implement, it has several limitations: 1) the reduced representation can cause information loss, 2) PCA can not be used for nonlinear data without further modification, 3) orthogonality of basis may not hold in some cases (~~*~~Ref. A Tutorial on Principal Component Analysis~~*~~).~~

### ~~Support vector machine~~

~~Support vector machine is a classification model which can be used to separate the data by transforming it into high dimensional space with the kernel technique. As the (~~*~~figure~~*~~) shows, certain data points are chosen as the support vector, and the area between the support vectors is called margin. SVM solves the classification problem by finding a hyperplane that can separate the data best and realize the maximal margin.~~

~~~~

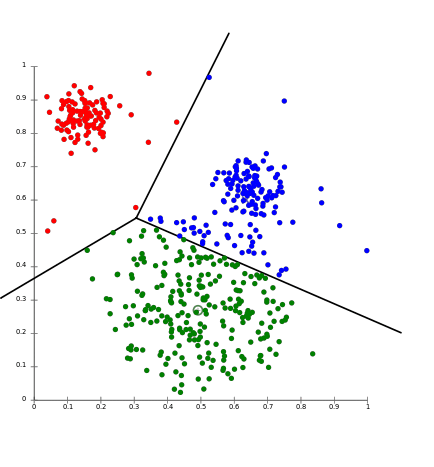
~~(~~*~~Ref. https://towardsdatascience.com/support-vector-machines-for-classification-fc7c1565e3~~*~~)~~

~~One fact about the classification task is that mapping the data into another space may significantly improve the performance. SVM takes advantage of this fact and solves the nonlinear separable issues using the kernel, transforming the data point into high- or infinite-dimensional space.~~

~~As a classification method, SVM works effectively in high-dimensional space. But it doesn't provide a probability estimate. Instead, it gives the classification result directly. Besides, the selection of kernels has a crucial impact on classification performance. Finally, from practical experience, the computation cost for SVM is much hight than other classifiers such as the random forest tree (~~*~~Ref. An introduction to support vector machines~~*~~).~~

### ~~Clustering~~

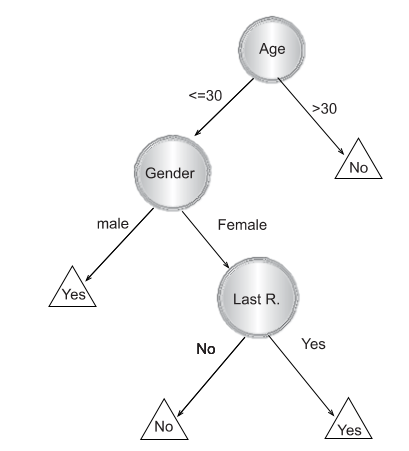
~~Clustering is a kind of unsupervised learning, and the goal is to assign data points into groups. A representative clustering algorithm is the K-means algorithm. K-means algorithm works by iterating between two steps until it reaches the termination criteria(~~*~~Ref. Top 10 algorithms in data mining~~*~~). In the initialization phase, several centroids need to be chosen first. In step 1, data points are assigned to different groups based on the distance to the centroids. Then in step 2, the centroids are recalculated based on the data assigned to its group. K-means iterated between those two steps until the algorithm converges. As (~~*~~fig~~*~~) shows, three centroids are chosen before the execution. After reaching the termination condition, all the data points are assigned into three groups. K-means algorithm is very sensitive to the initialization setting. Different centroids chosen at the initialization phase may lead to different clustering results. For TEP anomaly detection, clustering algorithms can be used for feature selection and improve detection performance.~~

~~~~

~~(~~*~~Ref.~~* [*~~https://aws.amazon.com/cn/blogs/machine-learning/k-means-clustering-with-amazon-sagemaker/~~*](https://aws.amazon.com/cn/blogs/machine-learning/k-means-clustering-with-amazon-sagemaker/)~~)~~

### ~~Decision tree~~

~~As a predictive model, the decision tree can be used for both classification and regression tasks. In this thesis, the decision tree is mainly used as a classifier to identify TEP abnormalities. The decision tree works by partition the instance space recursively and decides each node. As (~~*~~fig~~*~~) shows, a decision tree is consists of nodes and edges. The nodes without outgoing edges are called leaves or terminal nodes, and the nodes without incoming edges are called the root. Classification begins from the root node, after which each node splits the instance space into sub-spaces using a specific evaluation function until it reaches the leaves. The leaves represent the classification result, and edges correspond with certain sub-spaces.~~

~~~~

~~(~~*~~Ref. Data Mining with Decision Trees: Theory and Applications, chapter 1~~*~~)~~

~~The decision tree is an easy to explain machine learning algorithms because of the similarity to the human decision process. But it is not very robust and has high variance because a minor change in the training data can lead to a different decision tree construction. To further improve the decision tree's performance, the random forest tree is proposed (~~*~~Ref. Random Forests Leo Breiman 2001~~*~~).~~

### ~~Random forest tree~~

~~Random forest tree is also a predictive model based on the decision tree for classification or regression. Since the decision tree has poor generalization performance, (~~*~~Ref. The Random Subspace Method for Constructing Decision Forests~~*~~), (~~*~~Ref. Random Forests Leo Breiman 2001~~*~~) proposed~~ ~~random sub-space selection and bagging techniques, which leads to the invention of the random forest tree. The main idea of bagging is to construct many decision trees and make predictions using the~~ *~~majority vote~~* ~~(~~*~~Ref. An Introduction To Statistical Learning with Applications in R, chapter 8~~*~~). If the decision trees are not correlated after training, the majority vote of trees can decrease the variance. Random sub-space selection solves the problem between overfitting and high accuracy by selecting a random subset of features when splitting each decision tree's sub-space.~~

~~~~

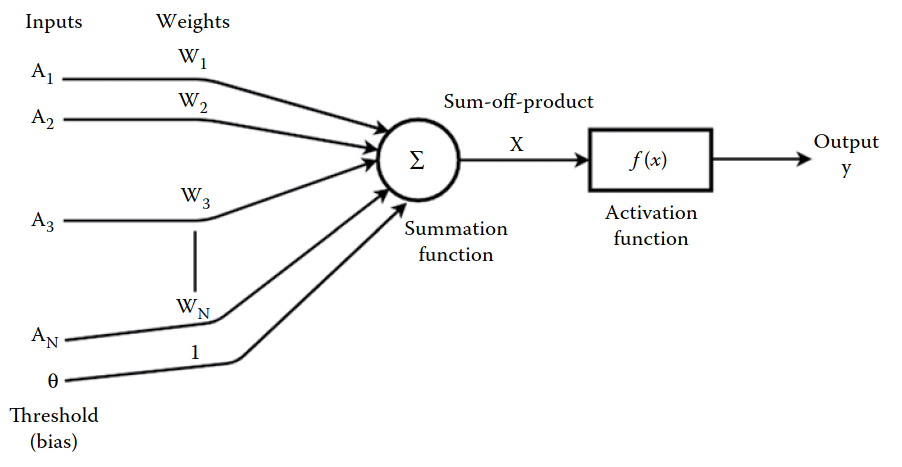
~~(~~*~~Ref. https://community.tibco.com/wiki/random-forest-template-tibco-spotfire~~*~~)~~

~~As (~~*~~fig~~*~~) shows, a random forest tree makes the classification based on the majority vote of Tree-1, Tree-2 and Tree-3. Since each tree is trained uncorrelated, the majority vote can help to provide better generalization and accuracy.~~

## ~~Deep learning~~

~~An artificial neural network (ANN) is inspired by studies on biological neurons and built with connected units, as (~~*~~fig~~*~~) shows. The input signals flow through the weights section, and the signal strength is adjusted according to the weight value. Then the summation function accumulates all the weighted signals and transmits them to an activation function. The bias signal has the function as an offset. A commonly used activation function is sigmoid function ranging from -1 to 1 and is defined by:~~

~~Stacking and connecting such units in different directions regarding the signal flows can build various artificial neural networks. (~~*~~Ref. Multilayer Feedforward Networks are Universal Approximators~~*~~) proved that an artificial neural network could be used as a universal approximator if enough units are in the middle layer. Based on the artificial neural network, deep learning uses multiple layers to solve complicated problems without the human's formal specification of knowledge. Besides, the representation of input data has a strong influence on machine learning performance. Deep learning solved the representation problem by dividing it into simple subproblems, i.e., the complexity of representations at different layers is different. This hierarchy working principle helps deep learning learn more abstract representations and achieve great success (~~*~~Ref. Deep Learning by Ian Goodfellow, Yoshua Bengio, Aaron Courville~~*~~).~~

~~~~

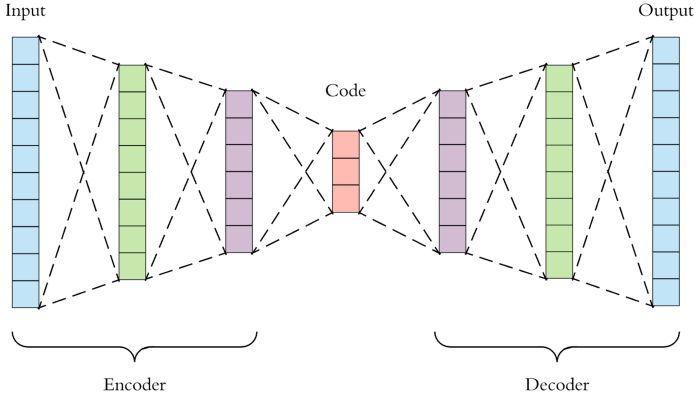
~~(~~*~~Ref. Machine Learning and Iot A Biological Perspective by Shampa Sen, Leonid Datta, Sayak Mitra~~*~~)~~

~~In the deep neural network history, hard to train is the main obstacle for the development. When the deep neural network is trained with gradient-based learning methods by backpropagation, the gradient may become unstable and lead to gradient exploding or vanish problems (~~*~~Ref. Neural Networks and Deep Learning - Michael Nielsen~~*~~). (~~*~~Ref. Gradient-based learning applied to document recognition~~*~~), (~~*~~Ref. Greedy Layer-Wise Training of Deep Networks~~*~~) and (~~*~~Ref. A Fast Learning Algorithm for Deep Belief Nets~~*~~) proposed several new approaches such as greedy layer-wise training and global training to solve that problem. Applying activation such as ReLU (Ref. Deep Sparse Rectifier Neural Networks) or residual networks (~~*~~Ref. Deep Residual Learning for Image Recognition~~*~~) can also ease the training problem. The event that promotes deep learning development is the work (~~*~~Ref. ImageNet Classiﬁcation with Deep Convolutional Neural Networks~~*~~) which achieves a new score for the image classification task. Besides image classification, deep learning also makes significant progress in speech recognition (~~*~~Ref. Deep Speech 2- End-to-End Speech Recognition in English and Mandarin~~*~~) and natural language processing (~~*~~Ref. Nematus: a Toolkit for Neural Machine Translation~~*~~).~~

### ~~Autoencoder~~

~~An autoencoder is a type of feed-forward artificial neural network which learns to reconstruct the input data. The autoencoder is consist of two components: an encoder which transforms the input to a latent space and a decoder that reconstruct the input . By minimizing the squared error between input and the reconstruction , where~~

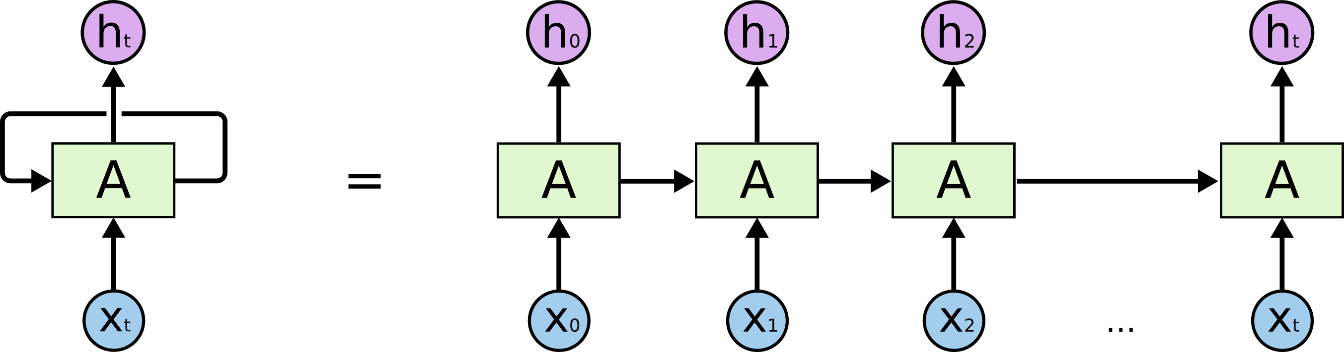
~~the autoencoder learns to the representation of the input data. The structure of autoencoder is as (~~*~~fig~~*~~) shows, the input is first compressed in by the encoder until the~~ *~~Code~~* ~~hidden layer, then expended by the decoder. The~~ *~~bottleneck~~* ~~design forces the autoencoder unable to copy the input precisely. The reason is that the~~ *~~Code~~* ~~hidden layer helps to learn the useful representation of input data (~~*~~Ref. Deep Learning by Ian Goodfellow, Yoshua Bengio, Aaron Courville~~*~~). Those representations can be further used for machine learning such as classification.~~

~~~~

~~(~~*~~Ref. https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798~~*~~)~~

### ~~Recurrent neural network~~

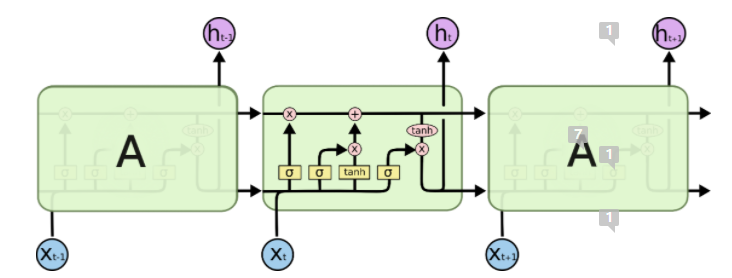
~~A recurrent neural network (RNN) is a class of artificial neural networks that can learn the mapping between input sequences and output sequences with the internal state. As (~~*~~fig~~*~~) shows, RNN uses loops to preserve previous computation's state information and process the input of next time sequence. One observation of the structure of RNN is that it should be able to build a connection between previous information and the current task. However, RNN can exhibit forgetting behavior which limits its representation capability when dealing with long-range sequences (~~*~~Ref. Local Feedback Multi-Layered Networks~~*~~). Besides, RNN suffers from training problems when long-term memory is involved in the task(~~*~~Ref. Learning Long-Term Dependencies with Gradient Descent is Difficult~~*~~), (~~*~~Ref. The Utility Driven Dynamic Error Propagation Network~~*~~). To mitigate the drawbacks of RNN, (~~*~~Ref. LONG SHORT-TERM MEMORY~~*~~) introduced Long Short-Term Memory~~

~~~~

~~(~~*~~Ref.~~* [*~~http://colah.github.io/posts/2015-08-Understanding-LSTMs/~~*](http://colah.github.io/posts/2015-08-Understanding-LSTMs/)~~)~~

### ~~Long Short-Term Memory Networks (LSTMs)~~

~~To solve the problem encountered in RNN, (~~*~~Ref. LONG SHORT-TERM MEMORY, 1997~~*~~) introduced Long Short-Term Memory network, an efficient gradient-based method using multiplicative gate units to control the error flow. (~~*~~Ref. Learning to forget: continual prediction with LSTM~~*~~) enhance the performance of LSTM by introducing an adaptive~~ *~~forget gate~~*~~, which enables LSTM to reset at an appropriate frequency. The LSTM network is composed of memory cell and every memory cell has weights and gates as figure shows.~~

~~~~

~~(~~*~~figure will be revised later~~*~~)~~

### Generative vs. Discriminative Modeling

In statistical classification problems, for a series of independent training samples , , , and . is a d-dimensional data while is a class label. A new data is categorized to class when it’s posterior probability (Bayes rule)

achieves the highest value. is a parametric model with parameter . Machine learning classification approaches can be classified into two major types: generative or discriminative modeling. The difference between the two models is the estimation of parameter .

Generative classifier estiminates the parameter using the maximum likelihood estimation (MLE) method:

,

.

(??? MLE needs to be introduced before)

Discriminative classifier estiminates the parameter parameter by maximing the conditional log-likelihood

,

(??? conditional log-likelihood needs to be introduced before)

By mathematical transformation, the can also be expressed as

,

where the first term is the generative objective function (*Ref. The Tradeoff Between Generative and Discriminative Classifiers*).

The generative models attempt to estimate the joint distribution of both input and output , which could also be used to generate synthetic data similar to existing data. The discriminative models learn to compute the mapping between input and output directly, i.e., calculate the conditional probability of when given (*Ref. Machine Learning Discriminative and Generative by Tony Jebara,* *Chapter 2*). In other words, the generative model focuses on understanding the data composition and basic characteristics of classes, whereas the discriminative models are designed to find the boundary between different classes (*Ref. Generative versus discriminative classiﬁers for android anomaly-based detection system using system calls ﬁltering and abstraction process*).

In general, generative models have following advantages:

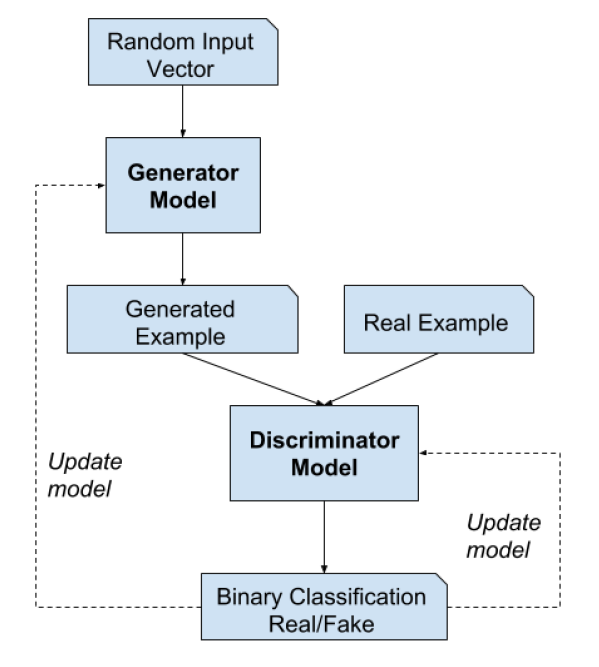
1. When the data is missing or partially labelled, generative model can compensate for missing data or augmente original labels.
2. New class of data can be learned independently by generative models without affecting previous learned classes.
3. Generative models can learn compositional features without taking samples of all combinations.

In contract to generative modesl, discriminative models have advantages such as better prediction performance because they learn the mapping between input and label directly (*Ref. Generative versus discriminative methods for object recognition*).

Recently, the success of image classification (*Ref. ImageNet Classification with Deep Convolutional Neural Networks*) which applied discriminative models has achieved huge success and it seems to imply that discriminative models have better performance than generative model in classification task. And a famous quote from Vladmir Vapnik “When solving a problem of interest, do not solve a more general problem as an intermediate step.” also implies that discriminative models are a preferred method. However (*Ref. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes*) showed that as the training examples increased, generative models which have reached asymptotic error can achieve better performance. (*Ref. Where’s Wally Now? Deep Generative and Discriminative Embeddings for Novelty Detection*) compared the discriminative and generative performance by developing a framework. Besides, their experiment demonstrated that when less information is available, the generative model can still realize close performance as discriminative methods.(*Ref. Generative versus discriminative methods for object recognition*), (*Ref. Comparison of Generative and Discriminative Techniques for Object Detection and Classiﬁcation*) showed that for object detection task, generative models can achieve higher accuracy while discriminative models are faster at prediction once trained. And the combination of both models is expected to achieve better performance. (*Ref. Anomaly Detection Combining Discriminative and Generative Models*) overcomes the shortcoming of discriminative models which suffer from imbalanced data by combining both generative and discriminative models.

### Generative Adversary Networks (GANs)

Generative adversary networks proposed in (*Ref. Generative Adversarial Nets*) is a framework to train generative model and avoid the intractable problem of probabilistic compuatation such as standard marginalization and conditioning operations (*Ref. A Fast Learning Algorithm for Deep Belief Nets*). GAN is composed of a generator and a discriminator. The generator is used to generate synthetic data while the discriminator attemps to classify data as real (from problem domain) or fake (generated by the generator). GAN works in a game theoretic scenario, i.e., the generator and discriminator learn by competing with each other (*Ref. Deep Learning*). As figure shows, the generator generates plausible example from a random input domain. Both the generated examples and real examples are provided to the discriminator to be identified as real or fake examples. The generator learns how to generate more *realistic* examples based on the classification result of the discriminator, i.e., the generator is penalized with large model parameter updates when the discriminator successfully identifies the real or generated examples. If the discriminator failed, the generator won't update the model parameter but the discriminator is penalized to update its parameters (*Ref. Generative Adversarial Networks with Python*).



(*figure will be revised later*)

Assume both the generator and discriminator are multilayer perceptrons (MLP need to be introduced before). For generator , the random input space is defined as and the mapping to data space is defined as , where are the model parameters. For discriminator , outputs a probability representing data came from the real dataset, where are ’s model parameters. In the training of GAN, is trained to maximize the probability to correctly assign labels for data from both the real dataset and . Meanwhile, is trained to minimize , i.e., minimize the probability that succeeds. Namely, and compete with each other in the minimax game of function :

,

where is the expectation.

**The training process.** In practice, the model parameters of and are updated iteratively. For example, is trained for steps followed by one-step training of . As figure (*Ref*.), from (a) to (d) shows the updating of discriminator (blue, dashed line) and generator (green, solid line) in more intuitive way. The generator and discriminator are trained simultaneously in order to win in the minmax game. After properaly training interations, the generator and discriminator will reach a point where discriminator can’t distinguish the distribution between generator and real data as (d).

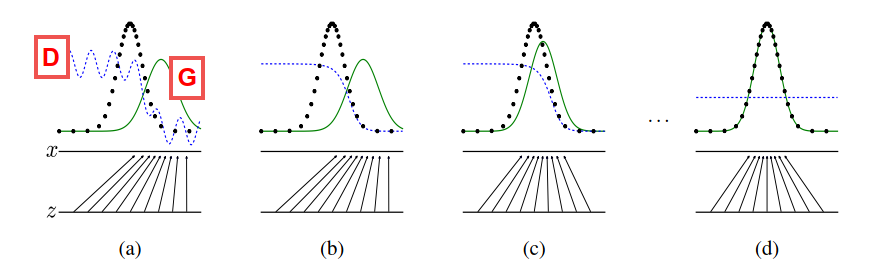


Figure : discriminator (blue, dashed line), generator (green, solid line), real data distribution (black, dotted line), (*figure will be revised later*) (*Ref. Generative Adversarial Nets*)

Some practical issues remain unsolved in the GAN training. At the early phase of training, if learns to distinguish real/fake examples faster than learns to generate more realistic fake examples. This situation causes to fail because the loss function can’t provide sufficient gradient information for . Therefore, (*Ref. Generative Adversarial Nets*) suggest to train to maximize to obtain stronger gradient information.

As an approach to develop generative modeling using deep learning methods, GAN has attracted attention from both practitioner and scholars. It can be applied to various scenarios. For example, it can be used for missing data imputation (*Ref. GAIN: Missing Data Imputation using Generative Adversarial Nets*), time-series prediction (*Ref. Time-series Generative Adversarial Networks*) and anomaly detection (*Ref. TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks*). This thesis focuses on analysis the application of GAN in anomaly detection and the comparison with other anomaly detection methods.

### Conditional GAN and Bidirectional GAN

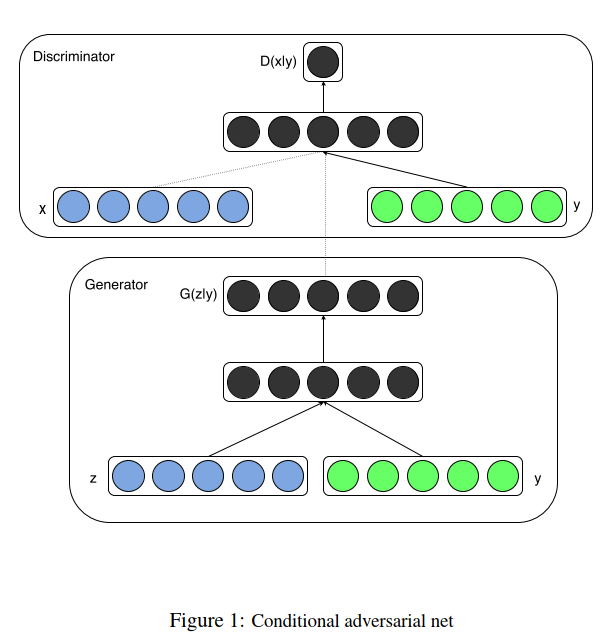
Conditional and bidirectional GAN are two innovative extensions of GANs framework, which provide the cornerstone for applying GAN in anomaly detection. Conditional GAN helps to build generative model to conditionally generate an output by feeding both from latent space and some extra information as input to generator and discriminator. The extra information can be class label, the style of images and so on. Bidirectional GAN extend the GAN by adding the *inverse mapping* which maps from real data space back to latent space with an encoder (*Ref. A Survey on GANs for Anomaly Detection*).

**Conditional GAN.** The application of GAN framework (*Ref. Unsupervised representation learning with deep convolutional generative adversarial networks*) can help to train generative model to generate images. However, there is no way to control the style or type of the generated images.

Conditional GAN (cGAN) first proposed in (*Ref. Conditional Generative Adversarial Nets*) proposed a way to conditional generating images by feeding the generator with a class label. Comparing to the GAN (*Ref. Generative Adversarial Nets*), the objective function is

,

where is the auxiliary information such as class labels. As figures shows, during the training, the class labels are first encoded as one-hot vectors and concatenated with the input from MNIST handwritten digit dataset, then fed as input to discriminator. The input for generator is similar to discriminator which is the concatenation of sampled from latent space and the class labels. To generate a new handwritten digit, feed the generator with input conditioned with label information.



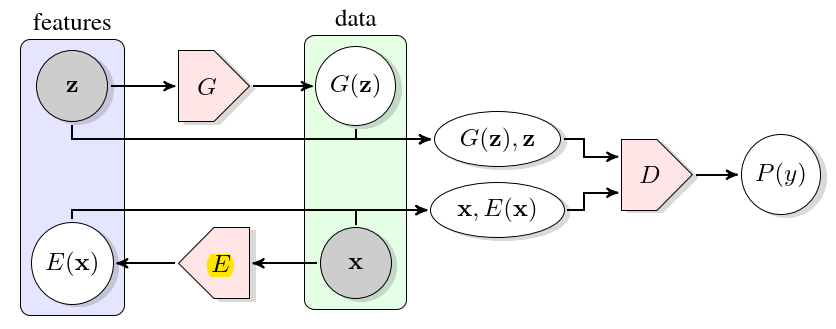
(*figure will be revised later*) (*Ref. Conditional Generative Adversarial Nets*)

(*Ref. Conditional generative adversarial nets for convolutional face generation*), (*Ref. Conditional Image Synthesis with Auxiliary Classiﬁer GANs*) and (*Ref. Image-to-Image Translation with Conditional Adversarial Networks*) had demonstrated the great potential of cGAN in image synthesis. (*Ref. Conditional Image Synthesis with Auxiliary Classiﬁer GANs*) also argued that the generated image can achieve better discriminability than the models which generate lower resolution images.

**Bidirectional GAN.** As an extension of GAN, (*Ref. Adversarial Feature Learning*) first proposed Bidirectional Generative Adversarial Networks (BiGANs) which can learn the inverse mapping (from data space to latent space: ). Their framework includes an encoder which learns the inverse mapping. With the additional encoder, BiGANs can be further extended for unsupervised feature learning task such as anomaly detection (*Ref. GANomaly Semi-Supervised Anomaly Detection via Adversarial Training, 2018*), (*Ref. Combining GANs and AutoEncoders for Efﬁcient Anomaly Detection*). The objective function of BiGAN for training as follows:

where

Comparing to GAN (*Ref. Generative Adversarial Nets*), BiGAN use the pairs such as or as the input for discriminator. As figure shows, sampled from latent feature space is provided to to generate fake sample pair for discriminator to classify. On the other hand, sample pair force discriminator to identify real samples. The output of is the classification probability value, where suppose to be 1 if input is real sample pair as .



(*figure will be revised later*) (*Ref. Adversarial Feature Learning*)

## Evaluation Metric

In this section, we introduce the evaluation methods to assess the performance of different anomaly detection algorithms.

### Confusion Matrix

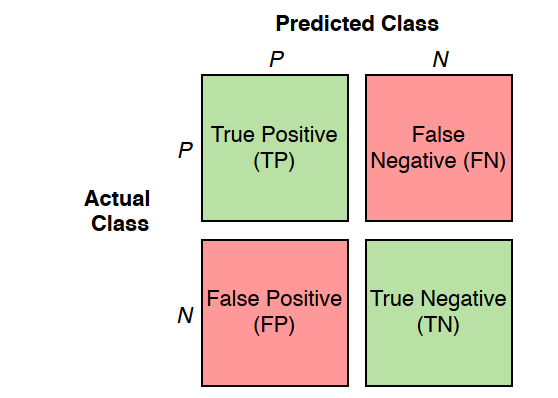
Confusion matrix are a commonly used evaluation method in binary classification where the predictions of classifier are presented in a contingency table with four cells. We first define true positives, false positives, true negatives and false negatives.

**True positive (TP).** Number of data points which are correctly predicted as the positive class.

**False positives (FP).** Number of data points which are incorrectly predicted as the positive class.

**True negatives (TN).** Number of data points which are correctly predicted as the negative class.

**False negatives (FN).** Number of data points which are incorrectly predicted as the negative class.



(*figure will be revised later*)

One assumption we made is that the result of predicting and assessing is either positive or negative. As figure shows, the green cells represent the correct predictions whereas the red cells are the incorrect predictions. Confusion matrix provides a visualization of the prediction performance of algorithms and we can further derive other evaluation metrics based on it.

### Precision

Precision measure the proportion of predicted positive points which are correctly positive points.

Precision is defined:

As can be seen from the formula , precision only considers positive cases (predictions and examples) and completely ignores the negative cases.

### Recall

Recall is the proportion of correctly predicted positive points in actual positive points.

Recall is defined:

### Accuracy

Accuracy measures the fraction of correctly predicted points.

Accuracy is defined:

### F1 Score

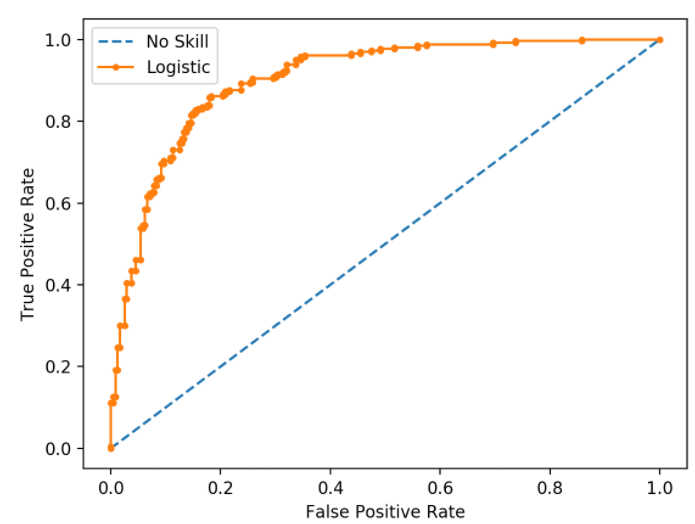
F1 score is defined:

As the formula shows, F1 score is calculate from the harmonic means of precision and recall which provides a balancing between them. F1 score reaches highest value of 1.0 when either precision or recall is 0.

### Receiver Operating Characteristics

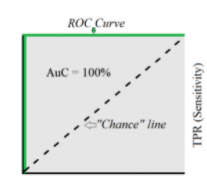
A receiver operating characteristics (ROC) is a graph which evaluates the performance of machine learning algorithms (*Ref. Signal detection theory Valuable tools for Evaluating Inductive Learning*). As figure shows, the ROC graph plots the (true positive rate) in axis and (false positive rate) in axis, where

At the lower left point , the classifier predicts not only no false positive points but also no true positives points. At the points the classifier predicts the points as positive unconditionally. The diagonal line (blue dotted line) connecting point and represent the performance of classifiers which make predictions by randomly guessing. For example, if a classifier predicts a point to be positive with probability 30%, it is expected to achieve a 30% TPR but also a 30% FPR. Therefore, when the classifier performs better than random guessing, the points of and are plotted as orange dotted line which is above the diagonal line.



(*figure will be revised later*)

**Area under an ROC curve.** Althoughan ROC curve can provide a straightforward description of the performance of machine learning algorithms, It is more convenient to use a single scalar value to compare different algorithms. (*Ref. The use of the area under the ROC curve in the evaluation of machine learning algorithms*) proposed area under the ROC curve (AUC) which is the portion the area underneath ROC curve in a unit square. A random guessing classifier has AUC value of 0.5 and no realistic classifier has AUC value less than 0.5. A perfect classifier has AUC value close to 1 as figure shows.



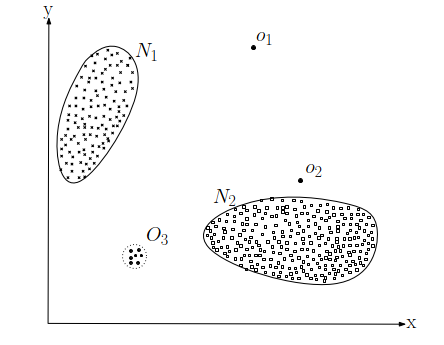
(*figure will be revised later*)

# Anomaly detection

Anomaly detection is one of the most important applications of data science. It refers to the problem of identifying the patterns of data which doesn’t satisfy expected behavior. Such patterns are also known as outliers, anomalies and exceptions, etc (*Ref. Anomaly Detection A Survey*). Another definition of anomalies as (*Ref. Identification of Outliers by D. M. Hawkins*) concluded anomalies as the data that deviates so much from the majority of the data and it was plausibly produced by a different mechanism. A straightforward method to detect anomalies is to define a normal region and identify the data which doesn’t belong to this region as anomalies. However, this method is not feasible in practical situation for several reasons (*Ref.. Anomaly Detection A Survey*): 1) There is often no clear and exact boundary between normal and abnormal regions; 2) The current definition of normal region might representative enough for detecting anomalies in the further. 3) The exact description of anomalies could be significantly different considering different application scenarios. For example, it is normal for an adult to gain 10% weight in summertime but abnormal to increase the height of 10%.

To mitigate those challenges (*Ref.. Anomaly Detection A Survey*) introduced a detailed classification of anomalies:

1. Point Anomalies: A data instance is categorized as point anomalies when it deviates significantly from the rest of the data. As figures shows, data , and locate far from the normal area and , therefore identified as point anomalies.



(*figure will be revised later*)

1. Contextual Anomalies
2. Collective Anomalies

分类需要重新划分

## Supervised anomaly detection

## Semi-Supervised anomaly detection

## Unsupervised anomaly detection

## GAN in Anomaly detection

**AnoGAN.** (*Ref. Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery*) first proposed Anomaly Detection with GAN (AnoGAN) to identify the location of anomaly in image. AnoGAN is a deep convolutional networks trained with only normal samples using GAN framework as (*Ref. Generative Adversarial Nets*). After the training, the generator has learned to generate normal samples rather than anomaly samples. The difference between the input and reconstruction can help to detect anomalies. As (*Ref. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*) discussed, the in latent space has the property of smooth transition. In other words, two close point , in latent space can generate similar images , in data space. In order to find the best (to fulfil ) in latent space of a query , AnoGAN use an iterative process by backpropagating in steps. In practice, they deployed a composed loss function to measure the anomality of the query , where

,

,

The residual loss measures the similarity between query sample and the generated sample , and the discriminator loss helps to improve the training stability of GAN. is the weight parameter between those 2 losses. The value of at step is also defined as anomaly score, where a small anomaly score implies that the query is similar to the normal samples during training, whereas a large anomaly score means that the query are anomaly sample. AnoGAN is the first proposed GAN framework showed that GAN can also be used for anomaly detection. The main limitation of AnoGAN is the calculation of anomaly score needs steps optimization, which is computational expensive (Ref. GANomaly Semi-Supervised Anomaly Detection via Adversarial Training).

**EGBAD.** To overcome the limitation of AnoGAN, (*Ref. Efficient GAN-Based Anomaly Detection*) proposed Efﬁcient GAN-Based Anomaly Detection (EGBAD), which is based on BiGAN. EGBAD adopt an additional encoder as in BiGAN to map points from data space to latent space. The encoder is jointly trained with generator, which avoid the optimization steps as in AnoGAN.

**GANomaly.** (*Ref. GANomaly Semi-Supervised Anomaly Detection via Adversarial Training*) introduced the GANomaly method which is inspired by BiGAN (*Ref. Adversarial Feature Learning*), AnoGAN (*Ref. Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery*) and EGBAD (*Ref. Efficient GAN-Based Anomaly Detection*). As figures shows, the network is composed of three sub-networks. The first sub-network is the generator part of the GAN model. The generator deploys a bottleneck autoencoder network with an encoder and decoder . The generator is first fed with from data space and deliver it to the encoder . The encoder transforms into point of latent space, where and it is also known as the compressed expression of autoencoder *(??? introduced in autoencoder part*). Then the decoder decodes into , where . The second sub-network is the encoder part of the GAN. The encoder transforms which is reconstructed by to the latent space point , where . The point and have the same dimensions. The third sub-network is the discriminator . The function of is to distinguish if the input pair are real or fake. GANomaly adopts the similar idea as using autoencoder for anomaly detection: if the generator is trained only with normal samples, it is not able to reconstruct abnormalities when the input is an abnormal sample. The objective function of GANomaly is composed of three loss functions:

,

,

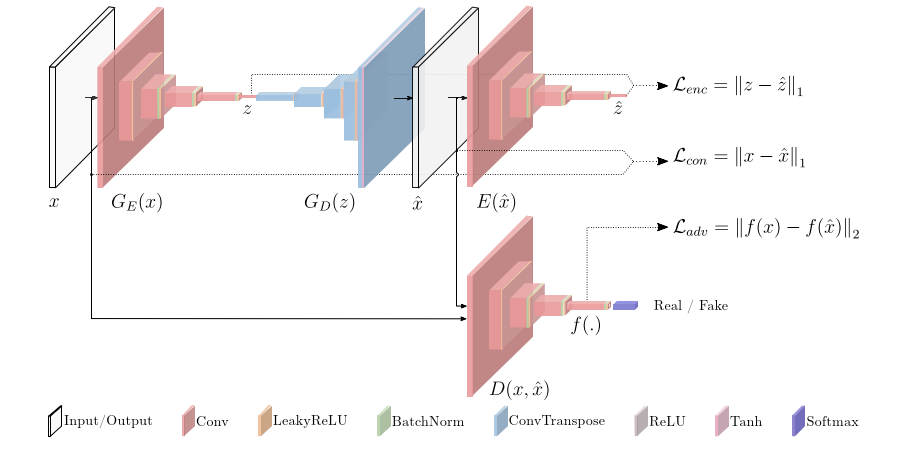
,

,

where , and are the weights of different loss functions. The adversarial loss is used to improve the stability of GAN training as (*Ref. Improved techniques for training gans*) recommended. is the intermediate layer’s output of discriminator . The contextual loss helps the Generator to reconstruct the input data as (*Ref. GANomaly Semi-Supervised Anomaly Detection via Adversarial Training*) discussed. Similar loss function is also used in (*Ref. Image-to-Image Translation with Conditional Adversarial Networks*) to help generator generate more realistic data. The encoder loss is employed to minimize the distance between points and in latent space, which can help to learn the feature representation in latent space. After the training, the detection of test sample is measured by anomaly score

,

where a higher anomaly score means the test point is likely to be anomaly points.



(*figure will be revised later*) (*Ref. GANomaly Semi-Supervised Anomaly Detection via Adversarial Training*)

**TadGAN.** (??? introduced in LSTM part)

## Comparison of different GANs for anomaly detection

## Summary