Contents

[1. Introduction 2](#_Toc71124660)

[1.1 Motivation 2](#_Toc71124661)

[~~1.2~~ ~~Problem Statement~~ 4](#_Toc71124662)

[~~1.3~~ ~~Research Questions~~ 4](#_Toc71124663)

[~~1.4~~ ~~Organization (may change)~~ 5](#_Toc71124664)

[~~2.~~ ~~Background Theory~~ 6](#_Toc71124665)

[~~2.1~~ ~~Machine Learning Concept~~ 6](#_Toc71124666)

[~~2.2~~ ~~Traditional algorithms~~ 8](#_Toc71124667)

[~~2.2.1~~ ~~Principal component analysis~~ 8](#_Toc71124668)

[~~2.2.2~~ ~~Support vector machine~~ 9](#_Toc71124669)

[~~2.2.3~~ ~~Clustering~~ 10](#_Toc71124670)

[~~2.2.4~~ ~~Decision tree~~ 11](#_Toc71124671)

[~~2.2.5~~ ~~Random forest tree~~ 12](#_Toc71124672)

[~~2.3~~ ~~Deep learning~~ 13](#_Toc71124673)

[~~2.4~~ ~~Deep Learning Training~~ 14](#_Toc71124674)

[~~2.5~~ ~~Deep Learning Performance~~ 14](#_Toc71124675)

[~~2.5.1~~ ~~Autoencoder~~ 15](#_Toc71124676)

[~~2.5.2~~ ~~Recurrent neural network~~ 16](#_Toc71124677)

[~~2.5.3~~ ~~Long Short-Term Memory Networks (LSTMs)~~ 17](#_Toc71124678)

[2.5.4 Generative vs. Discriminative Modeling 18](#_Toc71124679)

[2.5.5 Generative Adversary Networks 20](#_Toc71124680)

[2.5.6 Conditional GAN and Bidirectional GAN 22](#_Toc71124681)

[2.6 Evaluation Metric 25](#_Toc71124682)

[2.6.1 Confusion Matrix 25](#_Toc71124683)

[2.6.2 Precision 25](#_Toc71124684)

[2.6.3 Recall 26](#_Toc71124685)

[2.6.4 Accuracy 26](#_Toc71124686)

[2.6.5 F1 Score 26](#_Toc71124687)

[2.6.6 Receiver Operating Characteristics 26](#_Toc71124688)

[~~3.~~ ~~Methodology~~ 29](#_Toc71124689)

[~~3.1~~ ~~What are anomalies?~~ 29](#_Toc71124690)

[~~3.2~~ ~~Challenges in Anomaly Detection~~ 30](#_Toc71124691)

[~~3.3~~ ~~Output of Anomaly Detection Models~~ 30](#_Toc71124692)

[~~3.4~~ ~~Anomaly Detection Models~~ 31](#_Toc71124693)

[~~3.4.1~~ ~~Feature extraction anomaly detection framework~~ 31](#_Toc71124694)

[~~3.4.2~~ ~~Feature representation anomaly detection framework~~ 32](#_Toc71124695)

[~~3.4.3~~ ~~Unsupervised anomaly detection~~ 33](#_Toc71124696)

[~~3.4.4~~ ~~GAN in Anomaly detection~~ 34](#_Toc71124697)

[~~3.4.5~~ ~~Comparison of different GANs for anomaly detection~~ 36](#_Toc71124698)

[~~3.5~~ ~~Summary~~ 36](#_Toc71124699)

[~~4.~~ ~~Experiment~~ 37](#_Toc71124700)

[~~4.1~~ ~~What is TEP?~~ 37](#_Toc71124701)

[~~4.2~~ ~~What is Data Preparation/Pre-processing?~~ 38](#_Toc71124702)

[~~4.3~~ ~~Tennessee Eastman Process Simulation Data~~ 38](#_Toc71124703)

[~~4.3.1~~ ~~TEP Data Preparation~~ 40](#_Toc71124704)

[~~4.3.2~~ ~~TEP data exploration and visualization~~ 40](#_Toc71124705)

[~~4.4~~ ~~Baseline Models~~ 42](#_Toc71124706)

[~~4.5~~ ~~From reconstruction error to anomaly detection~~ 45](#_Toc71124707)

[~~4.6~~ ~~Autoencoder with multi hidden layers~~ 45](#_Toc71124708)

[~~4.7~~ ~~Autoencoder with multi hidden layers~~ 45](#_Toc71124709)

# Introduction

## Motivation

The machine learning (ML) area has undergone radical changes in the last several years. Academia and industry have invested billions of dollars in developing massively complex machine learning algorithms. As an essential subset of ML, Deep learning (DL) works by deploying multiple layers in the network and has 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 training data volume. In other words, deep learning models can continuously improve performance and outperforms traditional machine learning with more training data(*Ref. Deep Learning for Anomaly Detection A Survey, 2019*). Because of the overwhelming advantage of deep learning in big data processing, deep learning has been applied in various anomaly detection tasks such as surveillance video analysis (*Ref. Real-time illegal parking detection system based on deep learning*), cyber intrusion detection (*Ref. A deep learning approach for network intrusion detection system*) and medical diagnosis (*Ref. Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery*). This thesis focuses on evaluating different anomaly detection methods, especially deep learning methods, in the industrial process.

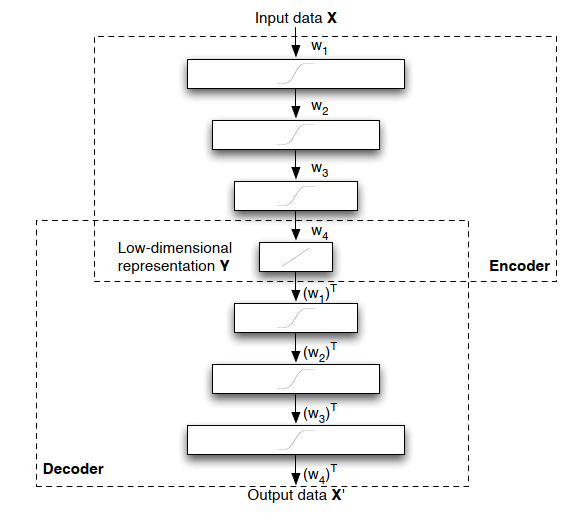
The industrial processes have gradually become highly complex to fulfill the requirement of high production efficiency and quality. More and more monitor and controllable variables are introduced, which pose challenges for system control. Early detection of anomalies and implementation of correspondence measures are critical for safety and effective operation. As discussed in [(*Ref. A survey of the application of basic data-driven and model-based methods in process monitoring and fault diagnosis*)], the anomaly detection 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 challenging to realize and economically unfeasible. 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. 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 may even completely break down some algorithms. These phenomena which not arise in low-dimensional data are also termed the *curse of dimensionality* (*Ref. Bellman, Richard Ernest; Rand Corporation (1957). Dynamic programming*). In (*Ref. A Few Useful Things to Know about Machine Learning*), Pedro M. Domingos gave two descriptive explanations for the *curse of dimensionality*. The first explanation is that the negative influence of noise accumulates as dimensionality increases, i.e., the influence of noise from irrelevant features may suppress the influence of relevant features. 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, methods such as feature selection (*Ref. Feature Engineering and Selection*) and feature extraction (*Ref. Foundations of Multidimensional and Metric Data Structures*) are proposed. 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*). In contrast, the feature extraction method extracts the features by transforming the high-dimensional data into a low-dimensional space. The feature extraction method can be divided into linear and nonlinear approaches. A representative linear feature extraction approach is principal component analysis (*Ref. A Tutorial on Principal Component Analysis*). The principal component analysis extracts the low-dimensional structures by representing the high-dimensional data into a linear subspace of lower dimensions. However, principal component analysis cannot adequately deal with nonlinear data like TEP data. To handle nonlinear high-dimensional data, autoencoder (aka. replicator neural networks) is proposed (*Ref. Outlier Detection Using Replicator Neural Networks, 2002*). As *fig* shows, an autoencoder consists of an encoder and a decoder part. The encoder part learns to represent the input data into a low-dimension space at the middle hidden layers (*bottleneck),* then the decoder part maps the data back to high-dimensional space. For autoencoders, it learns to copy the input approximately to its output. The encoder part of the autoencoder can be used to extract the data feature. One key point of this thesis is to explore the application of autoencoder in TEP's anomaly detection. Meanwhile, this thesis also seeks to provide a comprehensive comparison of machine learning methods regarding TEP data.



(*figure will be revised later*) (*Ref. Dimensionality Reduction: A Comparative Review*)

## ~~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, the goal of 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~~

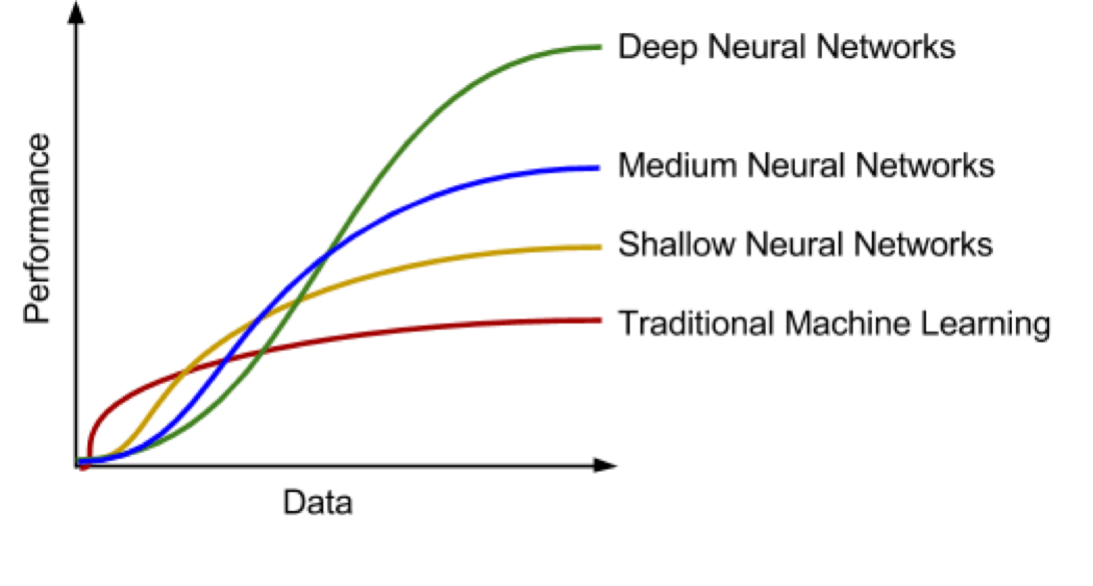
~~This chapter provides the necessary background and theoretical knowledge to understand the methodology in chapter 4. It first introduces the basic concepts of machine learning, then gives a broad overview of traditional machine learning and deep learning. Finally, it reviews the prior work and compares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

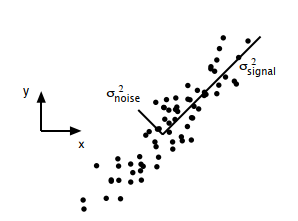
## ~~Traditional algorithms~~

~~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.~~

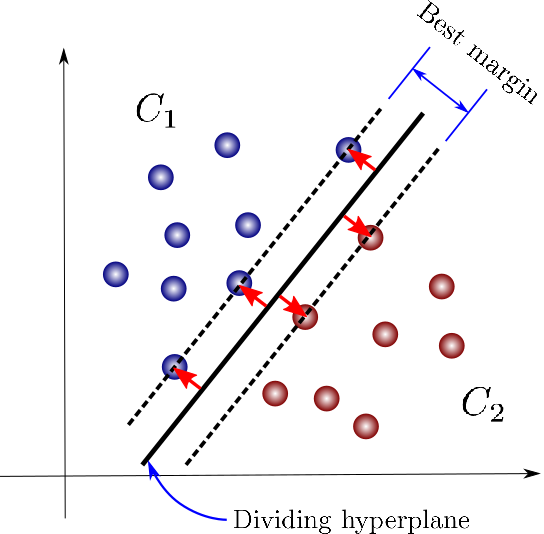
~~~~

~~(~~*~~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.~~

~~~~

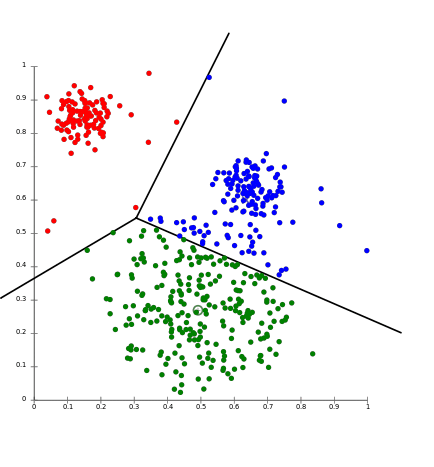
~~(~~*~~Ref. https://towardsdatascience.com/support-vector-machines-for-classification-fc7c1565e3~~*~~)~~

~~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~~

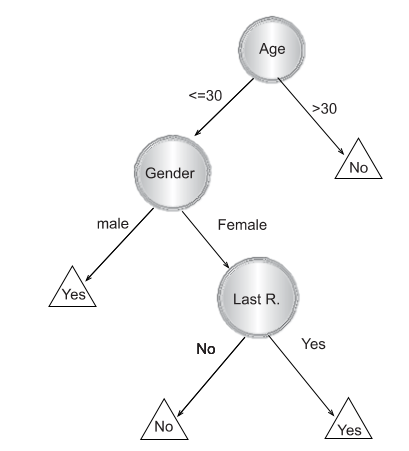
~~Clustering is a kind of unsupervised learning, and the goal is to assign data points into groups. A representative clustering algorithm is the K-means algorithm. K-means algorithm works by iterating between two steps until it reaches the termination criteria(~~*~~Ref. Top 10 algorithms in data mining~~*~~). 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.~~

~~~~

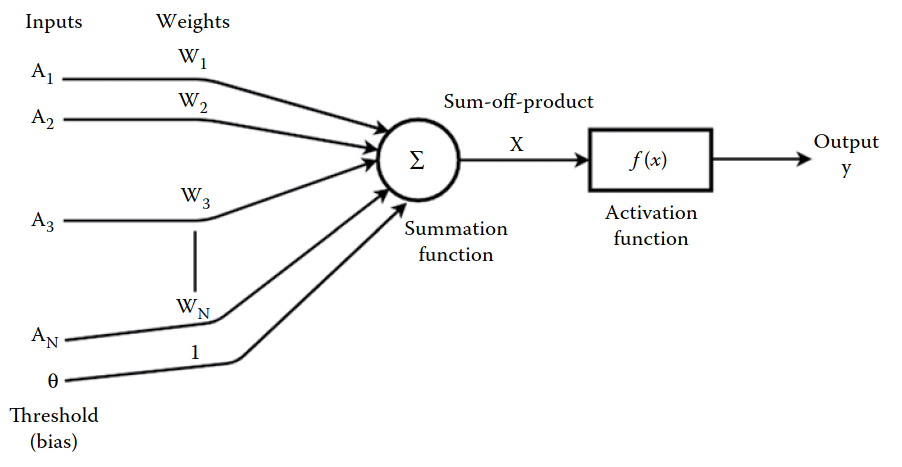
~~(~~*~~Ref. https://community.tibco.com/wiki/random-forest-template-tibco-spotfire~~*~~)~~

~~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~~*~~)~~

## ~~Deep Learning Training~~

**~~Backpropagation.~~**

~~We prove this problem NP-complete and thus demonstrate that learning in neural networks has no eﬃcient general solution.~~

**~~— Neural Network Design and the Complexity of Learning, 1988~~**

## ~~Deep Learning Performance~~

**~~Capacity~~**

**~~Regularization.~~**

**~~Overfitting:~~**

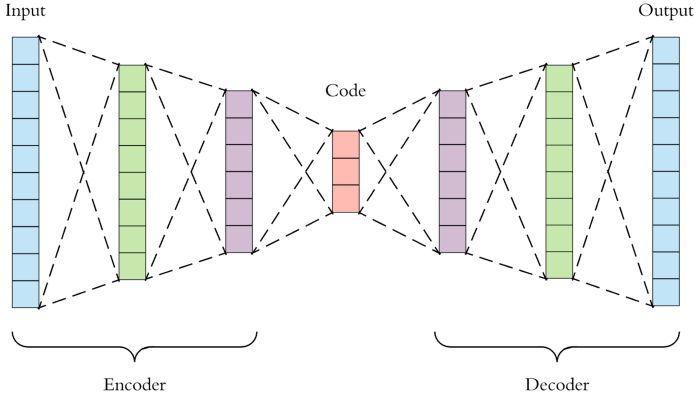
**~~Diagnostic Learning Curves不同的curve表型效果举例~~**

~~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~~

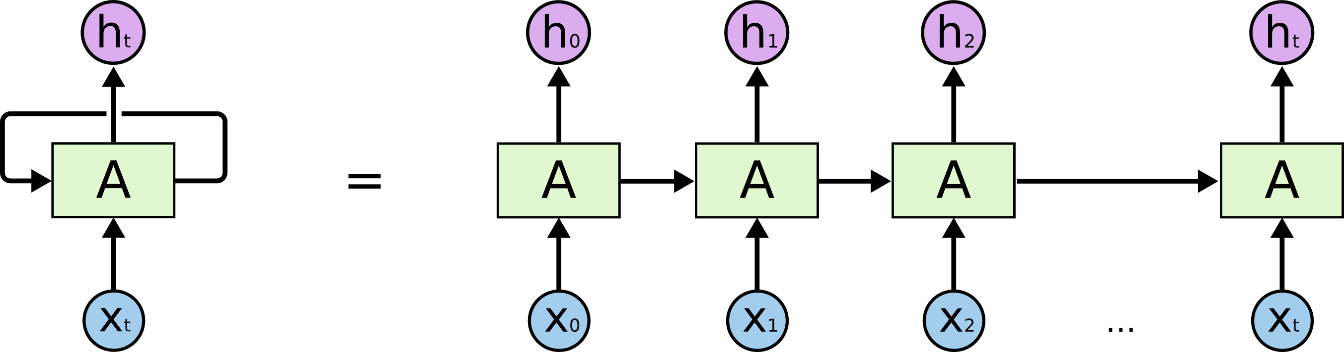
~~the autoencoder learns to the representation of the input data. The structure of autoencoder is as (~~*~~fig~~*~~) shows, the input is first compressed in by the encoder until the~~ *~~Code~~* ~~hidden layer, then expended by the decoder. The~~ *~~bottleneck~~* ~~design forces the autoencoder unable 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~~

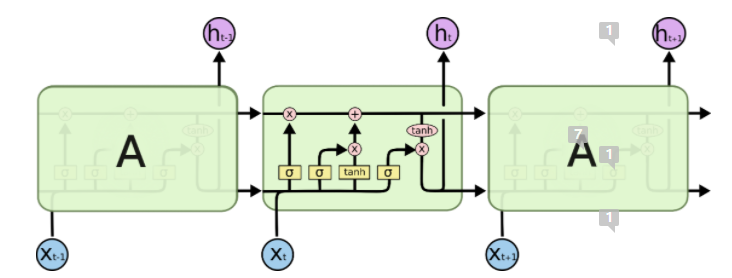
~~A recurrent neural network (RNN) is a class of artificial neural networks that can learn the mapping between input sequences and output sequences with the internal state. As (~~*~~fig~~*~~) shows, RNN uses loops to preserve previous computation's state information and process the input of next time 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

Generative and discriminative models are two important types of machine learning models.

For example, in statistical classification, given a series of independent training samples , , and . is a d-dimensional input data and is the corresponding output data (class label). We want to predict the class label of a new data based on the observations. According to the Bayes rule, a new data is categorized to class when the posterior probability (given ),

,

achieves the highest value.

is a parametric model that models the class-conditional probability and are the parameters of . are the multinomial distribution parameters of with and is the full parametrization of the joint density. The difference between the generative and discriminative models is the estimation of parameters .

Generative classifier estimates the parameters using the maximum likelihood estimation (MLE) method:

,

.

Discriminative classifier estimates the parameter by maximizing the conditional log-likelihood

,

.

By mathematical transformation, the relationship between and can also be expressed as

,

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

In general, generative models attempt to estimate the joint distribution of both input and output , generating synthetic data similar to observed data. Discriminative models learn to compute the mapping between input and output directly, i.e., calculate the conditional probability of given (*Ref. Machine Learning Discriminative and Generative by Tony Jebara,* *Chapter 2*). In other words, generative models focus on understanding the data composition and basic characteristics of classes. In contrast, the discriminative models are designed to find the decision boundary between data from different classes (*Ref. Generative versus discriminative classiﬁers for android anomaly-based detection system using system calls ﬁltering and abstraction process*).

In the practical application of machine learning algorithms, generative models have the following advantages (*Ref. Generative versus discriminative methods for object recognition*):

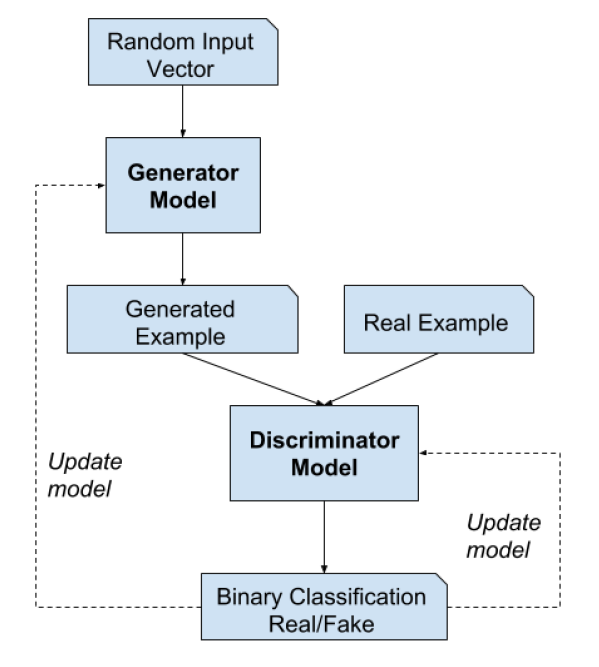
1. When the data is missing or partially labeled, generative models can compensate for missing data or augment original labels.
2. A new class of data can be learned independently by generative models without affecting the previously learned classes.
3. Generative models can learn compositional features without taking samples of all combinations.

Compared to generative models, discriminative models have advantages such as better prediction performance since they learn the mapping between input and label directly.

Recently, image classification (*Ref. ImageNet Classification with Deep Convolutional Neural Networks*) applied discriminative models has achieved huge success and seems to imply that discriminative models have better performance than generative models in the classification task. Besides, 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*) proved 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 models’ performance and demonstrated that when less information is available, the proposed generative model can still achieve comparable performance as the discriminative model. (*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 tasks, generative models could achieve higher accuracy while discriminative models are faster at prediction once trained. And the combination of both models is expected to achieve better performance. Further, (*Ref. Anomaly Detection Combining Discriminative and Generative Models*) overcame the shortcoming of discriminative models, which suffer from imbalanced data, by combining generative and discriminative models.

### Generative Adversary Networks

Generative adversary networks (GANs) proposed in (*Ref. Generative Adversarial Nets*) are neural network architectures composed of two neural networks: a generator and a discriminator. The generator is used to generate synthetic data, while the discriminator attempts to classify data as real (from data space) or fake (generated by the generator). GANs work in a game-theoretic scenario, i.e., the generator and discriminator learn by competing with each other (*Ref. Deep Learning*). As the *figure* shows, the generator generates plausible data examples 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 wouldn't update the model parameter, but the discriminator is penalized by updating its parameters (*Ref. Generative Adversarial Networks with Python*).



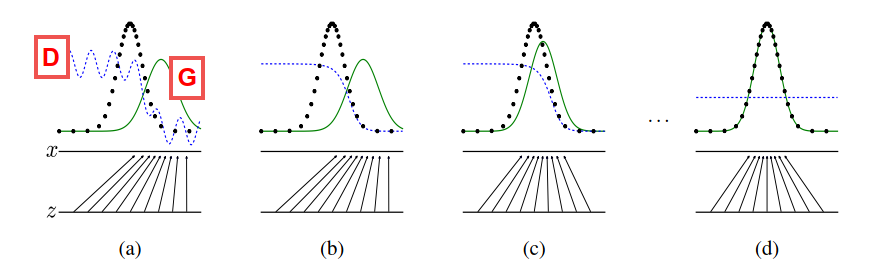
(*figure will be revised later*)

Assume both the generator and discriminator are multilayer neural networks. For generator , the random input space is defined as and the mapping to data space is defined as , where are parameters of generator . For discriminator , outputs a value representing the probability that data came from the real dataset, where are ’s parameters. In the training of GANs, discriminator is trained to maximize the ability to correctly assign labels for the data from both the real dataset or generated by generator . Meanwhile, generator is trained to minimize , i.e., minimize the probability that discriminator correctly classify the data. In other words, generator and discriminator compete with each other in the minimax game of function :

,

where is the expectation.

**The training process.** In practical implementation, the model parameters of and are updated iteratively. For example, is trained for steps followed b ya 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 a more intuitive way. The generator and discriminator are trained simultaneously to win in the min-max game. After adequately training epochs, the generator and discriminator will reach a state where the discriminator can’t distinguish if the data distribution is from the generator or real data as (d) shows.



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

However, some practical issues remain unsolved in the GAN training process. At the early phase of training, if the learning progress of (learns to distinguish real/fake examples) is faster than the progress of (learns to generate more realistic fake examples), G will fail to update the parameters. The reason that causes to fail is the loss function can’t provide sufficient gradient information for . Therefore, (*Ref. Generative Adversarial Nets*) suggest training to maximize rather than minimize to obtain stronger gradient information.

As a deep learning generative model, GAN has attracted attention from both practitioners and scholars. GANs avoid the intractable problem of probabilistic computation, such as standard marginalization and conditioning operations (*Ref. A Fast Learning Algorithm for Deep Belief Nets*). Besides, GANs can be applied to various scenarios such as 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 analyzes GANs' application in anomaly detection and the performance comparison of other anomaly detection methods.

### Conditional GAN and Bidirectional GAN

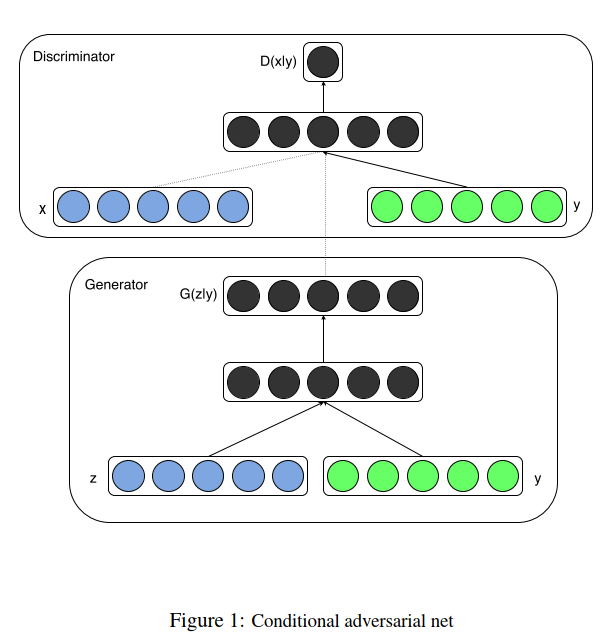
Conditional GAN and bidirectional GAN are two innovative extensions of the GANs framework, which provide the cornerstone for applying GANs in anomaly detection. Conditional GAN helps build a generative model to conditionally generate an output by feeding both from latent space and some additional information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

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

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

,

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 the discriminator . The input for the 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 the 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 a great potential of cGAN in image synthesis. (*Ref. Conditional Image Synthesis with Auxiliary Classiﬁer GANs*) also argued that the generated image could achieve better discriminability than the models that can only generate lower resolution images.

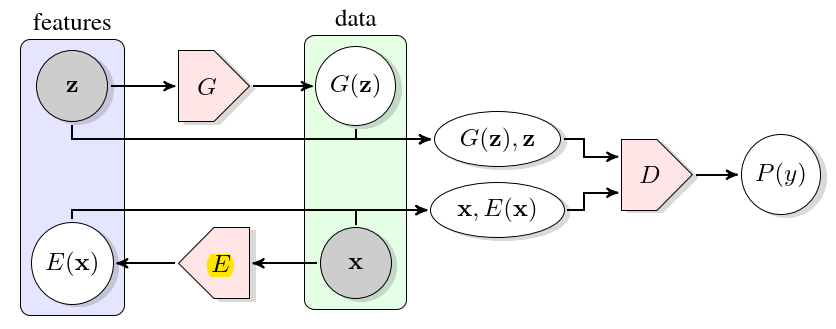
**Bidirectional GAN.** As an extension of GAN, (*Ref. Adversarial Feature Learning*) first proposed Bidirectional Generative Adversarial Networks (BiGANs), which can learn the inverse mapping (from data space to latent space: ). The BiGANs adopt an encoder subnetwork to map from latent space to data space. With the additional encoder, BiGANs can be further extended to unsupervised feature learning tasks 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 GANs (*Ref. Generative Adversarial Nets*), BiGAN use the pairs such as or as the input for the discriminator . As *figure* shows, sampled from latent feature space is provided to the generator to generate fake sample pair to train the discriminator to identify fake sample data. On the other hand, sample pair force discriminator to identify real sample data. The output of is the probability result of classification. is supposed to be close to 1 if the 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 anomaly detection algorithms.

### Confusion Matrix

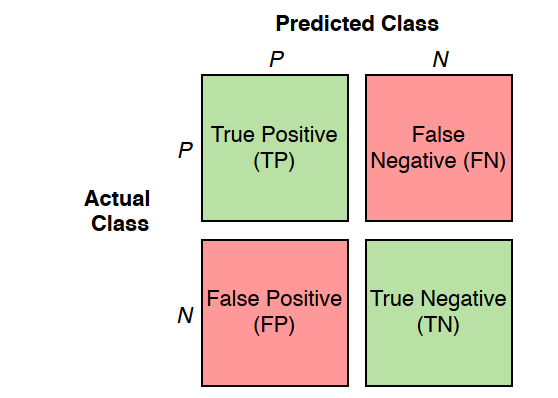
The confusion matrix is a commonly used evaluation method in binary classification tasks where the classifier's predictions are presented in a contingency table with four cells. We first define true positives, false positives, true negatives, and false negatives.

**True positive (TP).** The number of data points that are correctly predicted as the positive class.

**False positives (FP).** The number of data points that are incorrectly predicted as the positive class.

**True negatives (TN).** The number of data points that are correctly predicted as the negative class.

**False negatives (FN).** The number of data points that are incorrectly predicted as the negative class.



(*figure will be revised later*)

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

### Precision

Precision measures 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 instances and completely ignores the negative instances.

### 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

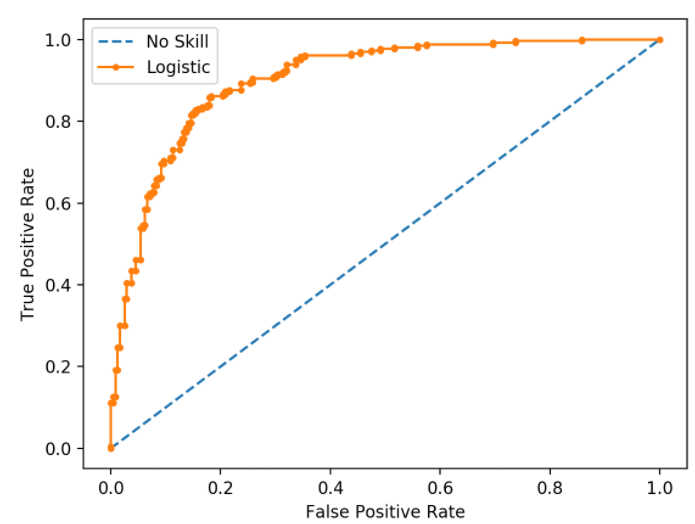
F1 score is defined:

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

### Receiver Operating Characteristics

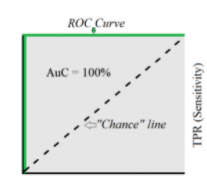
A receiver operating characteristics (ROC) is a graph tool that evaluates machine learning algorithms' performance (*Ref. Signal detection theory Valuable tools for Evaluating Inductive Learning*). As the *figure* shows, the ROC graph plots the (true positive rate) in axis and (false positive rate) in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

At the lower-left point , the classifier predicts no data instances as false positives or false positives. At the upper-right point , the classifier predicts the data instances as positive unconditionally. The diagonal line (blue dotted line) connecting both points and represents the classifiers that make predictions by randomly guessing. For example, if a random guessing classifier predicts the probability of a data instance being positive is 30%, it is expected to achieve a 30% and a 30% . Therefore, when the classifier performs better than a random guessing classifier, the points are connected as the orange dotted line above the diagonal line.



(*figure will be revised later*)

**Area under an ROC curve.** Althougha 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), the proportion of the area underneath the ROC curve in a unit square. A random guessing classifier has an AUC value of 0.5, and no realistic classifier should have an AUC value less than 0.5. A perfect classifier has an AUC value close to 1, as the *figure* shows.



(*figure will be revised later*)

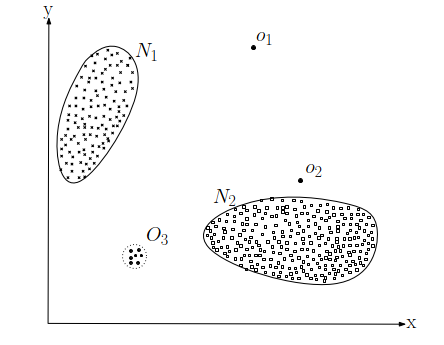
# ~~Methodology~~

~~This chapter first discusses the nature of input data and explains different types of anomalies. By comparing different types of anomaly data points, we points out the main challenges in anomaly detection. As anomaly detection is a fast envolving area in which numerous models are proposed and applied, the rest part of this chapter presents an overview of the methods in anomaly detection especially those involves deep learning.~~

## ~~What are anomalies?~~

~~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. Based on the nature and observation of anomaly data instances, (~~*~~Ref.. Anomaly Detection A Survey~~*~~) provided a detailed classification of anomalies:~~

1. ~~Point Anomalies: A data point 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. Point anomalies are the simplest and common type of anomaly which is also the focus point of most anomaly research.~~

~~~~

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

1. ~~Contextual Anomalies: A data point is termed as contextual anomalies (also conditional anomaly in (~~*~~Ref. Conditional Anomaly Detection, 2007~~*~~) when merely identified as anomaly in a user-specified area. For example, an 25 ambient temperature is recognized as abnormal situation in winter but considered as normal in summar.~~
2. ~~Collective Anomalies: A collection of series data is identified as anomaly with respect to the whole dataset. However, some individual points in the collection may not be anomaly.~~

## ~~Challenges in Anomaly Detection~~

~~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 not representative enough for detecting anomalies in the further as the decision boundary can be continually evolving. 3) The exact description or standard 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%.~~

~~In addition to the above challenges, anomaly detection also have following challenges (~~*~~Ref. Deep Learning for Anomaly Detection A Review~~*~~):~~

**~~High-dimensional data~~**~~. In the age of big data, the complexity and volume of data continuesly growing. Anomaly detection of time series data and multivariant data are two particularly challenging. For time series data, non-stationarity and dynamic behavior are two intractable problems. For multivariant data, high computational complexity and noise within the data affects the detecting performance greatly.~~

**~~Anomaly explanation~~**~~. For most safety-critical system, the failure or malfunction could lead to death or serious injury of people, severe property damage and environment damage (~~*~~Ref. Safety Critical Systems: Challenges and Directions~~*~~). However, the most anomaly dection model are deployed as black-box, in which only the input and outputs are visible. The internal function parameters are hard to build relationship with physical parameters such as temperature, pressure and volume (~~*~~Ref. Industrial control systemsimulation routines~~*~~). For anomalies data instances which are rare but safety-critical, a black-box model may predict biased result without further explanation of the result. A temporary solution is the inspection of anomaly detection result by human experts and manually tunning of the biased model. But this solution could decrease the effectiveness of the anomaly detection models.~~

## ~~Output of Anomaly Detection Models~~

~~The way to present the detected anomalies is critical since it involves the evaluation and application of anomaly detection methods. In general, the outputs of anomaly detection are anomaly scores or labels.~~

**~~Anomaly score~~**~~. To describe the level of deviation from normal operation state of anomly data instances, anomaly scores is applied. In (~~*~~Ref. Outlier Detection Using Replicator Neural Networks~~*~~) (~~*~~Ref. Outlier Detection with Autoencoder Ensembles~~*~~), the anomaly score (outlyingness) is defined as the reconstruction error over features of autoencoders. In (~~*~~Ref. Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery~~*~~), they defined the anomaly score as the combination of reconstruction error and discriminator output of the GAN.~~

**~~Labels~~**~~. Assigning category labels as the prediction of data instances is another way to present the anomaly detetion result. Labels can be the calculated from the output of a classifiers or by thresholding the reconstruction error of autoencoders. Generally, anomaly score provides more anomaly detection information than labels but not concise as labels.~~

## ~~Anomaly Detection Models~~

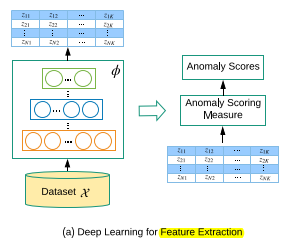
~~Considering the model structure, training framework, the anomaly detection methods is catorizied into three main categories: 1) Feature extraction anomaly detection, 2) Normal feature representation, 3) GAN in anomaly detection, 4)~~

### ~~Feature extraction anomaly detection framework~~

~~Feature extraction anomaly detection methods is consist of a feature extractor and an anomaly scorer ,~~

~~,~~

~~where is a feature extractor function with parameters , , and . In addition, can also be viewed as a function to reduce dimensionality. is the anomaly score with parameters and is the anamaly score. As figure shows, the feature extractor first extract features from dataset and transform it to extract features in latent space, then the extracted features is fed to the anomaly scorer to calculate the anomaly scorer. The feature extractor and anomaly scorer can be trained independently or jointly, which is also termed as mixed or fully deep approach in (~~*~~Ref. Deep One-Class Classification~~*~~) when employ deep learning methods. These two training methods lead to different objective functions. In mixed approach, is pretrained in a preceding step before leading the features into anomaly scorer. In other words, can be seen as a data preprocessing function for anomaly scorer and the anomaly scorer is later trained with the propocessed data. For fully deep approach, and are trained simutinously with the a single objective function.~~

~~~~

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

~~To reduce the data dimensionality, methods such as PCA (~~*~~Ref. An improved SVM integrated GS-PCA fault diagnosis approach of Tennessee Eastman process~~*~~) (~~*~~Ref. Nonlinear dynamic process monitoring based on dynamic kernel PCA~~*~~), autoencoders (~~*~~Ref. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning~~*~~) and encoder part of GAN (~~*~~Ref. f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks~~*~~) can be deployed. For the anomaly scorer , classifier such as SVM (Ref. SVM-based Deep Stacking Networks), random forest tree (~~*~~Ref. Random forests classifier for machine fault diagnosis~~*~~) or neural network classifier (~~*~~Ref. ???~~*~~) are popular methods. In general, Feature extraction anomaly detection works under the assumption that the extracted features can retain discriminative informations and there is a boundary which can separate data classes identify anomalies by anomaly scores.~~

~~Feature extraction anomaly detection framework has one significant advantages: A large amount of feature extractors and anomaly detector can be directly combined. It has two main drawbacks (~~*~~Ref. Deep Learning for Anomaly Detection A Survey~~*~~), i.e., performance relys heavily on accurate labels and balanced distribution of training data if the anomaly scorer works as a classifier. Accurate labels usually require huge amount of efforts of domain experts to manually assign. Furthermore, anomalies are rare events comparing to normal data instances which leads to imbalanced training data distribution between posivite class instances and negative class instances.~~

### ~~Feature representation anomaly detection framework~~

~~Feature representation anomaly detection refers to deploying the models such as autoencoder or other models with encoder-decoder architecture to detect anomlies. As discussed in chapter 2, an autoencoder must be restricted to copy the input to output only approximately. Otherwise the autoencoders can’t learn the most salient features of the data instances (~~*~~Ref. Deep Learning by Ian Goodfellow, Yoshua Bengio, Aaron Courville~~*~~). The definition is as follows:~~

~~,~~

~~,~~

~~,~~

~~,~~

~~where is the encoder with parameters , is the decoder with parameters . The objective function is to find the optimal parameters pair which minimize the reconstruction error. is the reconstruction error using the optimized . Generally, this type of anomaly detection model relys on the assumption, that the model can reconstruct the normal data instances from latent space (low dimensionality) better than anomaly data instances when it is only trained with normal data instances.~~

~~In practice, several variations of autoencoders have been proposed and achieved excellent performance.~~

**~~Stacking~~**~~.~~

**~~Denoising~~**~~.~~

**~~Sparse~~**~~.~~

**~~Variational autoencoder~~**~~.~~

**~~Contractive autoencoder~~**~~.~~

### ~~Unsupervised anomaly detection~~

~~Autoencoders are the core of all Unsupervised DAD models. These models assume a high prevalence of normal instances than abnormal data instances failing which would result in high false positive rate (~~*~~Ref. Deep Learning for Anomaly Detection A Survey, 2019.pdf~~*~~).~~

### ~~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:~~

~~,~~

~~,~~

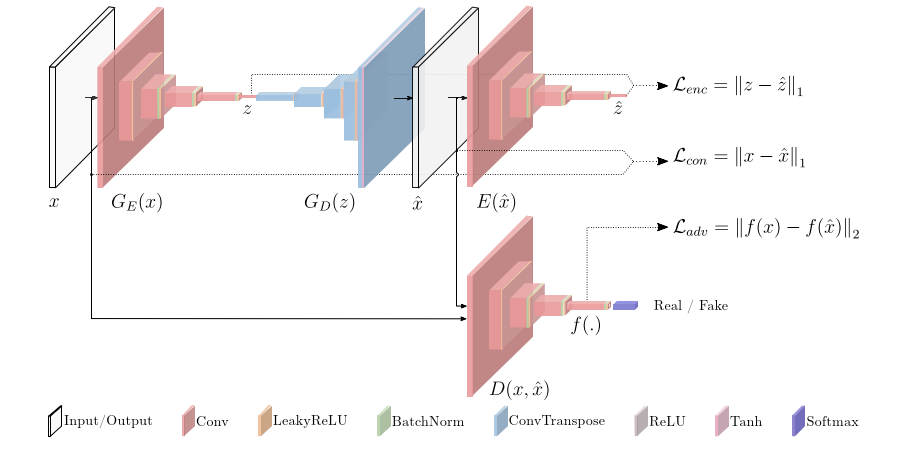
~~,~~

~~,~~

~~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~~

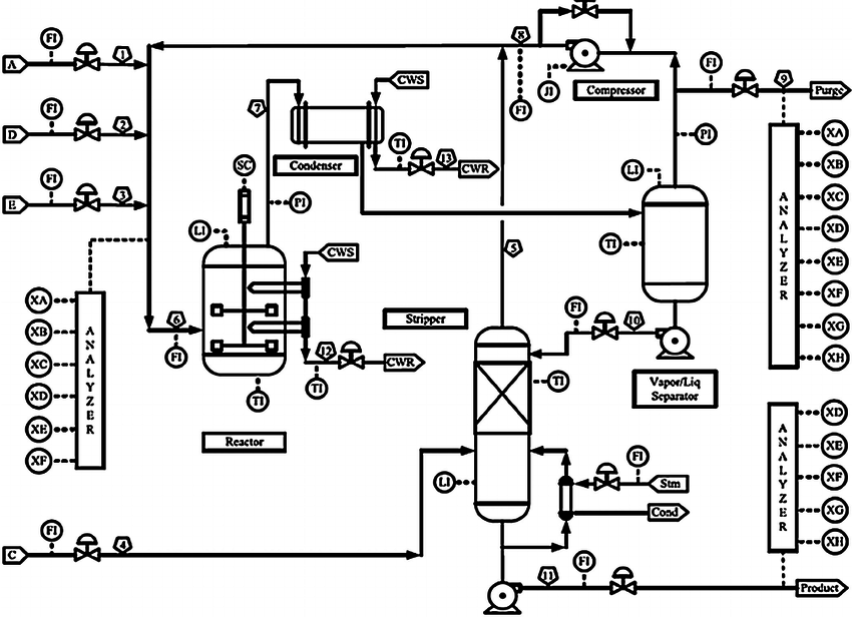
## ~~Summary~~

# ~~Experiment~~

~~In this chapter we first describe the data used for experiment and the basic steps for data preparation. Then we provide an extensive evaluation of both classical and deep learning anomaly detection methods. Finally, a performance comparasion and analysis among different methods is given.~~

## ~~What is TEP?~~

~~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~~*~~)~~

## ~~What is Data Preparation/Pre-processing?~~

~~Data preparation/pre-processing refers to operations such as addition, deletion and transformation of raw training data. Since the training of machine learning models relys heavily on data, incorrect data preparation can break model’s predictive performance (~~*~~Ref. Applied Predictive Modeling by Max Kuhn, Kjell Johnson, chapter 3~~*~~).~~

~~Although data preparation depends highly on the raw data, the machine learning model and underlying algorithsm, it can be summarized into following steps (~~*~~Ref. Data Preparation for Machine Learning by Jason Brownle, chapter 1~~*~~):~~

**~~Data cleaning~~**~~. There are various reasons that could cause the mistakes or incorrect values in data. The goal of data cleaning is to identify and correct them.~~

**~~Feature selection~~**~~. To address the curse of dimensionality (~~*~~introduced before~~*~~), one solution is to select the most relevant input features to the model’s output.~~

**~~Data transforms~~**~~. Generally refers to changing the distributions or types of raw data.~~

**~~Feature engineering~~**~~. Engineering new features based on training data may improve the model’s performance. However, the creation of new features may require experts’ knowledge.~~

**~~Dimensionality reduction~~**~~. Alternative to feature selection, dimensinality reduction refers to projecting the input data into a lower-dimensional space, meanwhile retaining the most meaningful properties of the original data.~~

## ~~Tennessee Eastman Process Simulation Data~~

~~This thesis uses the Tennessee Eastman Process (TEP) Simulation data from (~~*~~Ref. Additional Tennessee Eastman Process Simulation Data for Anomaly Detection Evaluation~~*~~) as the training and testing data in the experiments. The data is generated in the Tennessee Eastman Process Simulation and composed of four types of RData files:~~

**~~fault\_free\_training.RData~~**~~. Data generated in normal operating conditions for model training.~~

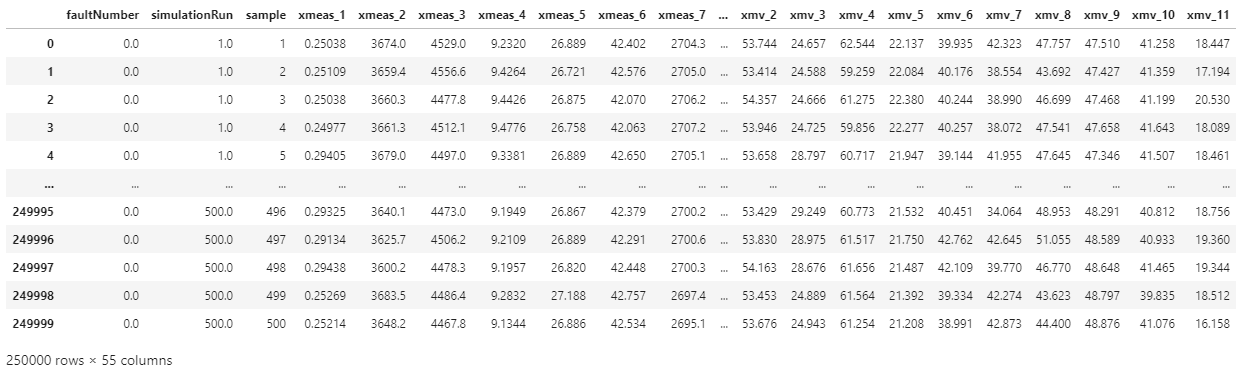
**~~fault\_free\_testing.RData~~**~~. Data generated in normal operating conditions for model testing~~

**~~faulty\_training.RData~~**~~. Data generated in abnormal operating conditions for model training.~~

**~~faulty\_testing.RData~~**~~. Data generated in abnormal operating conditions for model testing.~~

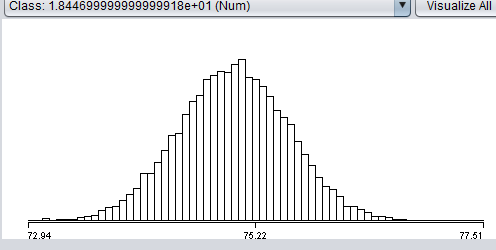
~~Each dataframe contains 55 columns as figure shows. The first column ist the faultNumber, which ranges from 0 to 20. 0 stands for fault free data, whereas 1-20 are the indexes for 20 types of fault. The second column ranges from 1 to 500 are different random number generator state generating the TEP dataset. The third column are the sample index in one simulaiton run. The TEP variables are sampled every 180 seconds. The left columns are the process variables: 41 measured and 12 manipulated variables.~~

~~For~~ **~~fault\_free\_training.RData~~** ~~and~~ **~~faulty\_training.RData~~**~~, the sample index in one simulation ranges from 1 to 500 with a duration of 25 hours. For~~ **~~fault\_free\_testing.RData~~** ~~and~~ **~~faulty\_testing.RData~~**~~, the sample index in one simualtion ranges from 1 to 960 with a duration of 48 hours. For~~ **~~faulty\_training.RData~~** ~~(resp.~~ **~~faulty\_testing.RData~~**~~), the fault is introduced at sample index 20 (resp. 160).~~

~~~~

~~(~~*~~figure will be revised later~~*~~) (fault\_free\_training.RData)~~

~~As the plot shows, the 7th variable measurement shows normal distribution. Almost all other 51 features had normal distributions.~~

~~~~

~~(~~*~~figure will be revised later~~*~~) (7th variable measurement distribution of fault\_free\_training.RData)~~

### ~~TEP Data Preparation~~

~~In the data cleaning step, there is no mistakes or errors found in the TEP data (~~*~~Ref. Additional Tennessee Eastman Process Simulation Data for Anomaly Detection Evaluation~~*~~). Data preparation steps such as feature selection, feature engineering and dimensionality reduction are not applied. Because we want to first build a baseline autoencoder model, which as discussed in (~~*~~Ref. Deep learning chapter 14~~*~~) can learn the useful properties of the data. In data transformation, the data is normalized with formula as recommended in (~~*~~Ref. AutoEncoder based High-Dimensional Data Fault Detection Detection System~~*~~):~~

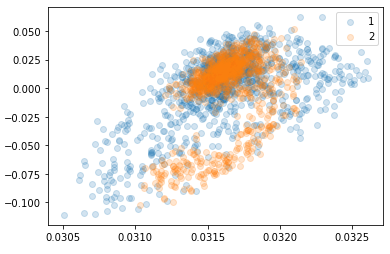
~~,~~

~~where and is the mean value and standard deviation of the value in -th column, refers to the sample index of data. is set to 2 because otherwise large part of data may fall outside of output range when we deploy as the activation function of baseline autoencoder. Both the and are first calculated based on~~ **~~fault\_free\_training.RData~~**~~, then used as the scaler for normalzitaion of other datasets.~~

### ~~TEP data exploration and visualization~~

~~Before feeding the training data to the machine learning model, we want to explore the high-diemensional data with visualization techniques to better understand the data distribution of different fault types.~~

**~~Visualization with PCA~~** ~~(~~*~~PCA is introduced before~~*~~). Using PCA to extract the first two principle component of faulty data with fault type 1 and 2. As the figure shows, there is no clear boundary that can separate the data of fault 1 and fault 2. Further, two types of data overlaps with each other. One reason is that PCA is a linear methods which can better separate dissimilar data instances in low-dimensional representation. However, for high-dimensional and nonlinear data like TEP data, it is more important to maintain the close distance between similar data instances (~~*~~Ref. Visualizing Data using t-SNE~~*~~).~~

~~~~

~~(~~*~~figure will be revised later~~*~~)(PCA visualization of data for fault 1 and fault 2)~~

**~~Visualization with t-SNE~~**~~. t-SNE is a visualization for high-dimensional data which is capable of preserving both the global and local structure of data in low-dimentsional space. Unlike PCA which separates data using the global structure, t-SNE focus on covert the data into low-dimentsional space using the pairwise similarities (local structure). Besides, t-SNE has series of tunnable parameters, “perplexity” and “steps” are two parameters which can influence the performance greatly. “perplexity” is plays the role of adjust the effective number of neighbors, in other words, it can balance the global and local structure of the data.~~

~~(~~*~~Ref. Visualizing data using t-SNE~~*~~).~~

**~~Visualization with variational autoencoder~~**~~. ???~~

**~~Visualization with LDA~~**~~. ???~~

~~数据数值范围分析~~

~~Scaling speeds learning because it helps to balance out the rate at which the weights connected to the input nodes learn.— Eﬃcient BackProp, 1998~~

~~Convergence is usually faster if the average of each input variable over the training set is close to zero. — Eﬃcient BackProp, 1998.~~

~~参考书~~

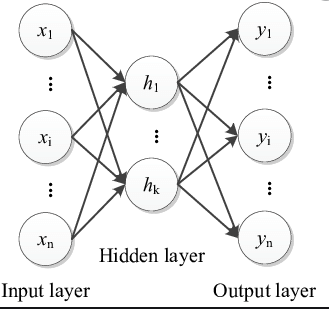
## ~~Baseline Models~~

~~Before build complex machine learning model, an autoencoder is first trained as the baseline model. An autoencoder is chosen as the baseline model because it is easy to train and can provide decent performance considering accuracy and inference speed as (~~*~~Ref. Outlier Detection Using Replicator Neural Networks, 2002~~*~~) discussed.~~

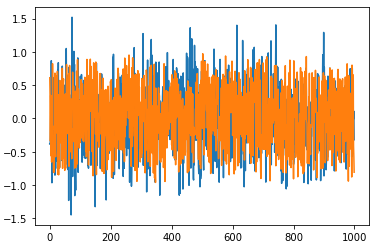
~~The baseline autoencoder is a feed-forware 3-layers perceptron with one hidden layer as figures shows. The autoencoder is trained with~~ **~~fault\_free\_training.RData~~** ~~and to reconstructed normal data. Mean square error is used to calculated the reconstruction error:~~

~~,~~

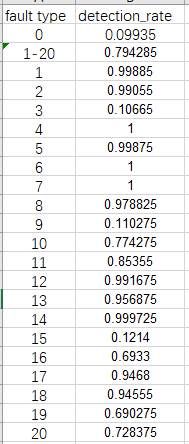
~~where is the input data and is the reconstructed data. A threshold is chosen to measure the reconstruction error.~~ **~~fault\_free\_testing.RData~~** ~~and~~ **~~faulty\_testing.RData~~** ~~are used to calculate the false alarm (false positive rate) and detection rate (true positive rate). Fault free data (resp. faulty data) with is classified as false positive (resp. true positive) data. In practice value is chose such that false positive rate is kept around 10% and early stopping is set to prevent overfitting.~~

***~~~~***

~~(~~*~~figure will be revised later~~*~~) (??? add layer node number)~~

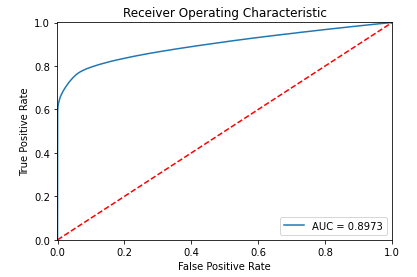
~~~~

~~(~~*~~figure will be revised later~~*~~) (reconstruction of features)~~

~~~~

~~(~~*~~figure will be revised later~~*~~) (detection rate of different fault types) (??? number accuracy revise)~~

~~As figure shows, the false alarm is 0.9935 and the detection rate is 0.794285. However the detection rate for fault 3, 9, 15 is less than 0.13. Those three types of fault is very difficult to detect.~~

~~~~

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

## ~~From reconstruction error to anomaly detection~~

## ~~Autoencoder with multi hidden layers~~

* ~~weight Regularization~~
* ~~Activity Regularization: 在act前,或后进行reg~~
* ~~Weight Constraints~~

## ~~Autoencoder with multi hidden layers~~