# Deep learning for Anomaly Detection: Theory and Applications

Jason J. Jung
Chung-Ang University
Seoul, Korea
j2jung@gmail.com
http://intelligent.pe.kr

# About me

- Jason J. Jung
- Professor at Chung-Ang University (Seoul, Korea)
- Around 200 articles on SCI journals (ESWA, INFFUS, INS, KBS, IPM, Plos One, Sci. Rep., J. Building Eng., etc)
- Editorial board members on INFFUS, JWS, and IPM
- Ex-Board member on NRF@KR
- Visiting professor at NII@JP, NTTU@VN, UM@MY

# About me

- Research question
  - How can AI help people to share their knowledge for collaboration?
- Knowledge representation with Ontologies and Knowledge graphs
- Collaboration network with social network analytics
- Recommendation & negotiation

- Data stream mining & anomaly detection

# Outline of talk

- Basic concept on anomaly detection
- Anomaly detection on multiple time series
- Applications and experiences
  - Traffic congestion detection
  - EEG
  - Climate change
- Open issues
  - Anomaly localization
  - Early detection

# Deep learning for Anomaly Detection: Theory and Applications

Jason J. Jung

Chung-Ang University

Seoul, Korea

j2jung@gmail.com

http://intelligent.pe.kr

# What is Anomaly?

- Anomalies
  - Deviated data points from expected data patterns

# What is anomaly detection?

- Anomaly detection is to identify the anomalous labels of data in a given dataset X.

- Question:
  - Can you detect the anomalies with a rule (e.g., fire alarm)?
- Answer:
  - Anomaly, if x (e.g., room temperature) > 80
  - Normal, otherwise

# Various applications

- Intrusion detection
  - Detecting unauthorized access in computer networks
  - Dataset: access log on the servers

# Various applications

- Fraud detection
  - Detecting fraudulent applications for financial organizations
  - Such as credit card, insurance claims, etc

  - Dataset: user transaction history

# Various applications

- Health care and medical diagnosis
  - Detecting disease on medical data
  - Such as cancer, heart attack, seizure, etc.

  - Dataset: medical images, eeg, ecg, and various EHR

# Various applications

- IIoT and Data stream monitoring
  - Detecting abnormal device and system behavior
  - Dataset: data streams from sensors

# Various applications

- Security and surveillance
  - Identifying abnormal scenes in video records (e.g., CCTV)
  - Dataset: video/image streams

# Various applications

- Autonomous vehicle development
  - Be aware of the road condition & scenes
  - Obstacles and pedestrians nearby vehicles

  - Dataset: video/image streams from camara or lidar (radar)

# Various applications

- Intrusion detection
- Fraud detection
- Health care and medical diagnosis
- IIoT and data stream monitoring
- Security and video surveillance
- Autonomous vehicle development
- And so on

# Challenges of Anomaly detection

- Can not take fully supervised approaches
  - Anomaly labels are very sparse, usually done by human expert manually
    - Fire alarms -> abnormal room  temp was decided by human experts.
  - Even with labels, the anomalies are rare. Normal and anomaly data are extremely imbalanced.
- Need to learn the high dimensional data patterns of normal data
- The notion of anomaly is subjective, varies from applications to applications
- The boundary between normal and anomalies is often not precise.

# Conventional algorithms

| Categories | Principles | Sub-categories | Popular AD techniques |
|---|---|---|---|
| Classification | Learn a discriminative boundary around the normal instances | Multi-class | |
| | | One-class | One-class SVM |
| Distance-Based | Define a distance measure to separate normal and abnormal data | Nearest Neighbor: distance to local neighborhood | LOF (local outlier factor), COF |
| | | Clustering: distance to the cluster belongs to | K-means, CBLOF |
| | | Projection-based: distance defined on a low dimensional space | PCA, Isolation Forest |
| Statistical Models | Normal data occur in high probability regions of a stochastic model | Parametric | Gaussian Mixture Model, Regression-Based (e.g. ARIMA) |
| | | Nonparametric | Kernel density estimator |

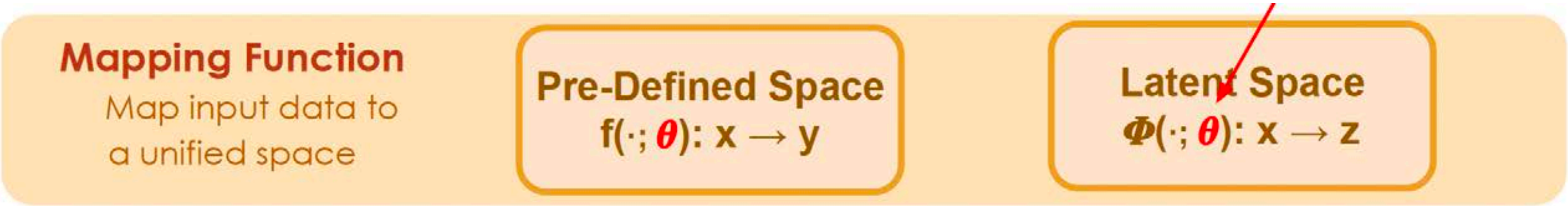
# Anomaly detection algorithms

- Generalized formulation

- Learning data representation
- Detecting anomalies

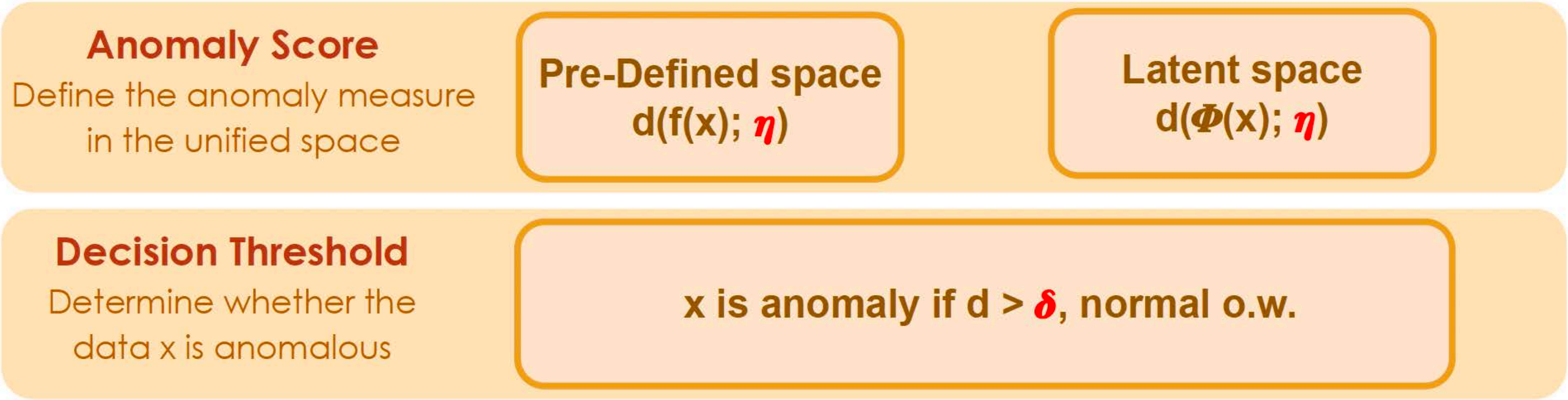
# Learning data representation

- Mapping function
  - Map input data to a unified space



# Detecting anomalies

- Define the anomaly measure in the unified space



# Anomaly detection algorithm

| | | |
|---|---|---|
| **Mapping Function**<br>Map input data to a unified space | **Pre-Defined Space**<br>$f(\cdot; \boldsymbol{\theta}): x \rightarrow y$ | **Latent Space**<br>$\boldsymbol{\Phi}(\cdot; \boldsymbol{\theta}): x \rightarrow z$ |
| **Anomaly Score**<br>Define the anomaly measure in the unified space | **Pre-Defined space**<br>$d(f(x); \boldsymbol{\eta})$ | **Latent space**<br>$d(\boldsymbol{\Phi}(x); \boldsymbol{\eta})$ |
| **Decision Threshold**<br>Determine whether the data x is anomalous | **x is anomaly if $d > \boldsymbol{\delta}$, normal o.w.** | |

# Model estimation

- How to estimate the following parameters

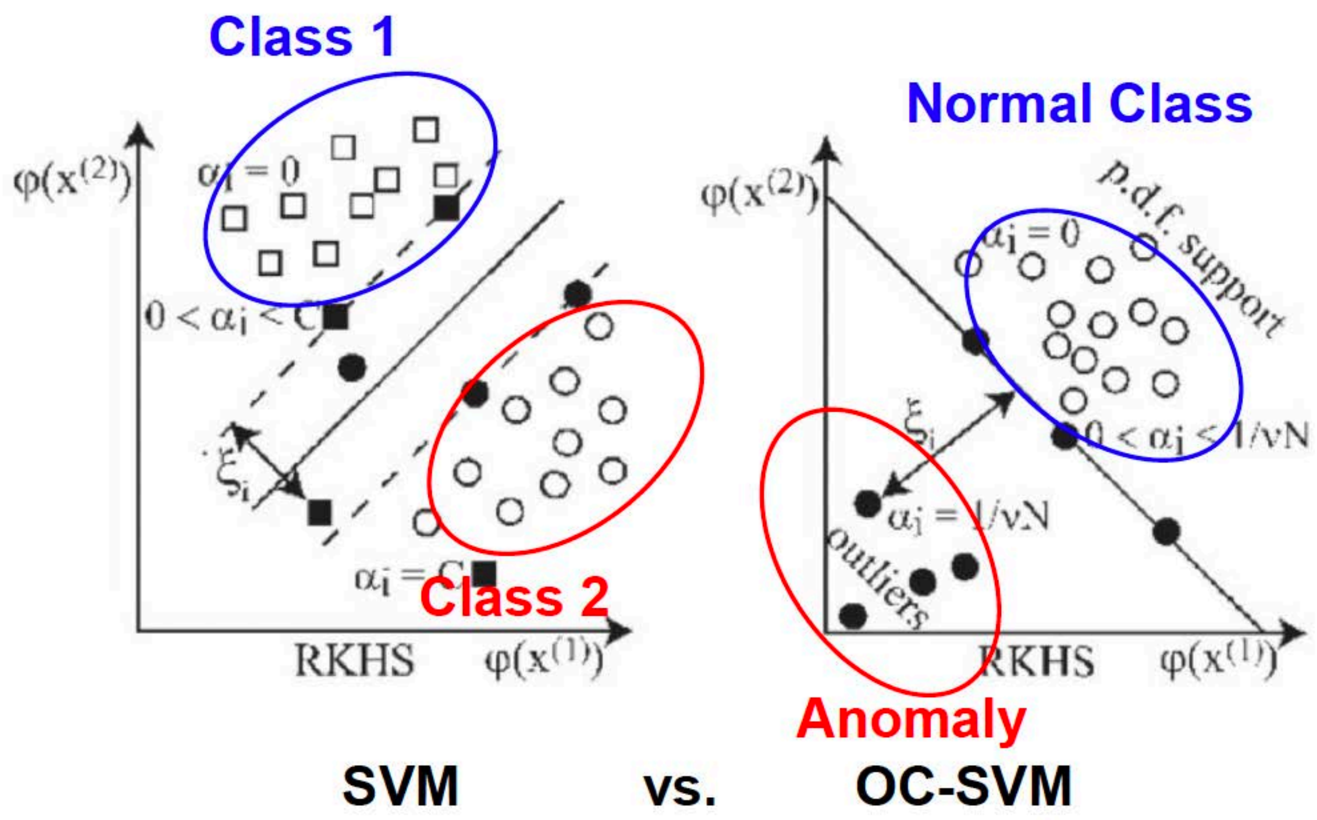
$$\theta, \eta, \delta$$

- w/ labeling
  - Supervised approach
- w/o labeling
  - Unsupervised approach
- w/ sparse labeling
  - Semi-supervised approach

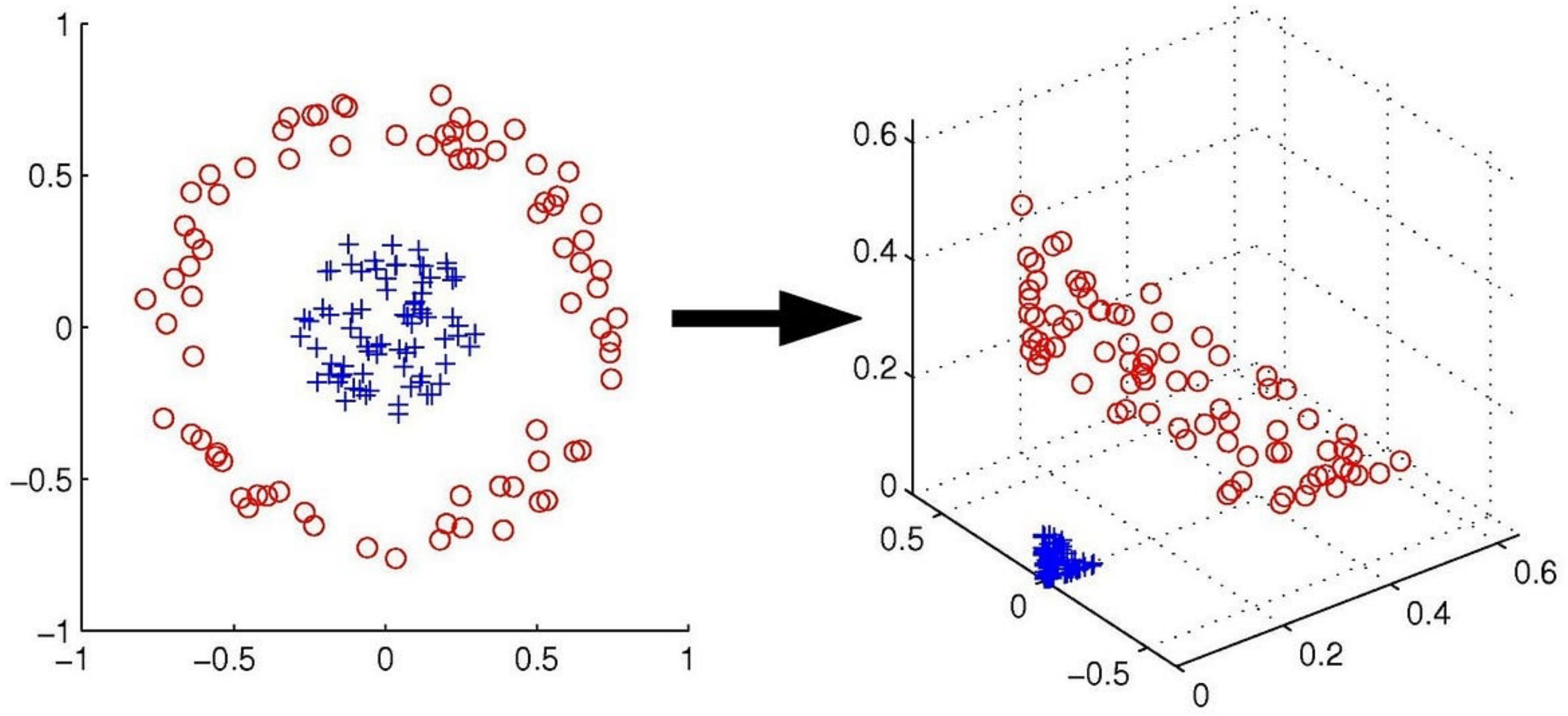
# Anomaly detection by algorithms

- Marginal approach
  - SVM -> One class classification
- Distance-based approach
  - KNN (or K-means)
- Statistical approach
  - Statistical distributions
  - Dynamic linear model

# SVM for One Class Classification



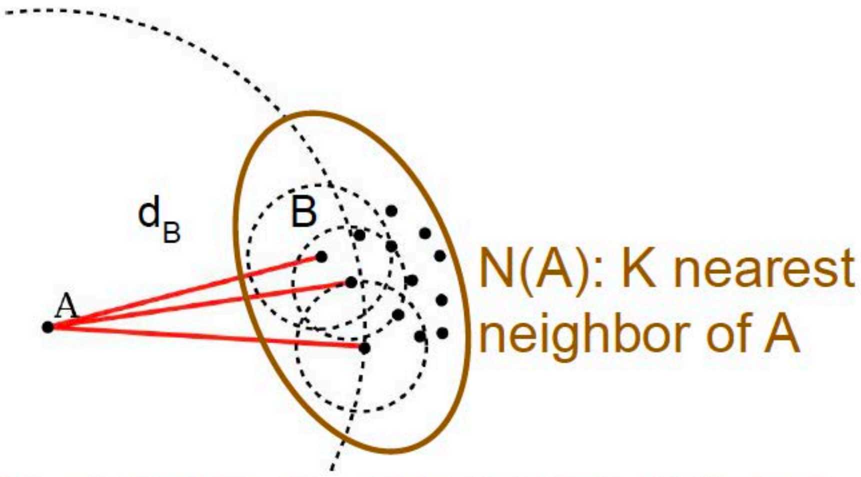
# SVM for One Class Classification



# SVM for One Class Classification

| | |
|---|---|
| **Mapping Function**<br>Map input data to a unified space | **Latent Space**<br>**Kernel Function $\varphi(x)$** |
| **Anomaly Score**<br>Define the anomaly measure in the unified space | **Distance to the origin**<br>$d(x) = w \cdot \varphi(x)$ |
| **Decision Threshold**<br>Determine whether the data is anomalous | **x is anomaly if $d(x) < \rho$, normal o.w.** |

# K-NN for Local Outlier Factor



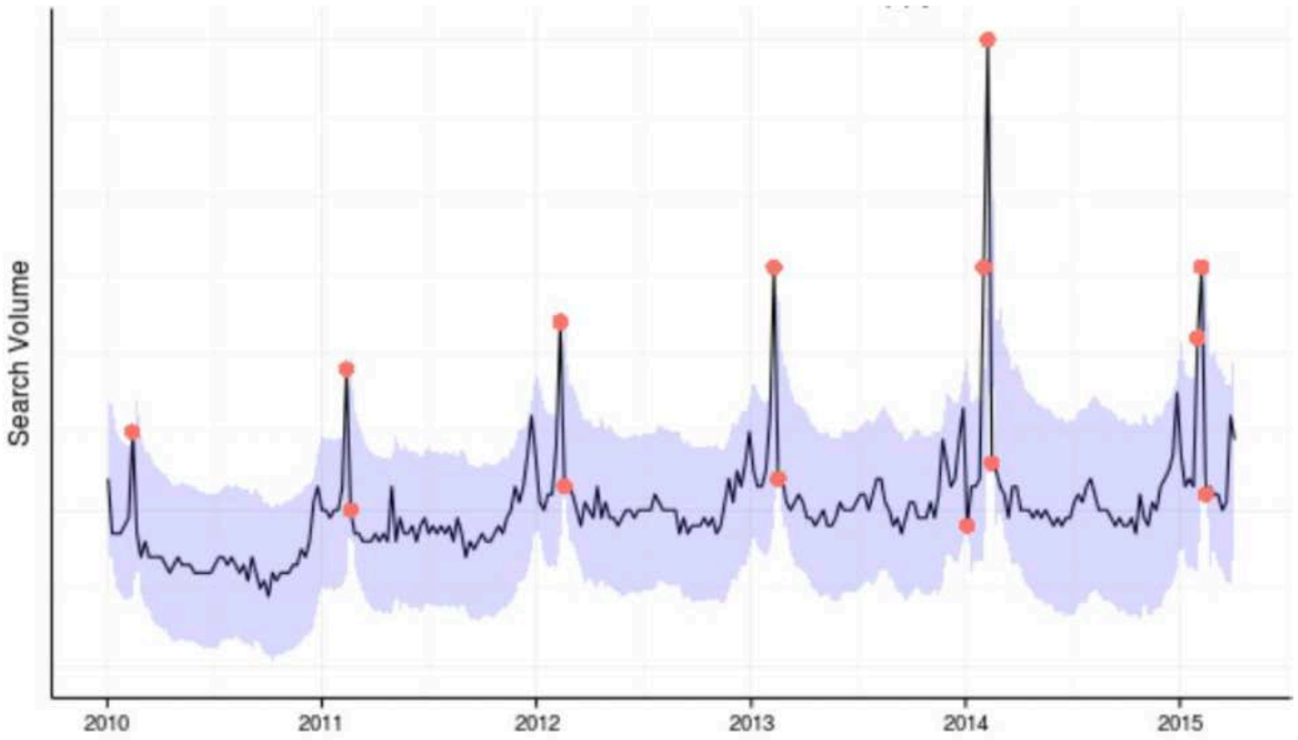

Source: https://en.wikipedia.org/wiki/Local_outlier_factor

$$\text{local-dist}(A) : ld(A) = \sum_{B \in N(A)} \frac{d_B}{|N(A)|}$$

# K-NN for Local Outlier Factor

| | |
|---|---|
| **Mapping Function**<br>Map input data to a unified space | **No Mapping** |
| **Anomaly Score**<br>Define the anomaly measure in the unified space | $\mathrm{LOF(x)} = \mathrm{ld(x)} / \sum_{B \in N(x)} \mathrm{ld(B)} / \lvert N(x) \rvert$ |
| **Decision Threshold**<br>Determine whether the data is anomalous | x is anomaly if $\mathrm{LOF}_k\,(x) > 1$, normal o.w. |

# Dynamic linear model with statistical distributions




Source: https://jgeer.com/anomaly-detection-how-to-analyze-your-predictable-data/

# Dynamic linear model with statistical distributions

| | |
|---|---|
| **Mapping Function**<br>Map input data to a unified space | **Pre-Defined Space**<br>$f(\cdot;\, \theta_t, V_t, W_t): x_{1:t-1} \rightarrow x_t$ |
| **Anomaly Score**<br>Define the anomaly measure in the unified space | **Prediction Error**<br>$d(x_t) = \| f(x_{1:t-1}) - x_t \|$ |
| **Decision Threshold**<br>Determine whether the data is anomalous | $x_t$ is anomaly if $d(x_t) > \delta$, normal o.w.,<br>$\delta$ comes from test statistics or fine-tuned with anomaly labels |

# Challenges & Opportunities

- Learn better nonlinear and hierarchical discriminative features from data
- Capture complex and high-dimensional data structures, including those with dependency
- Generic model architecture suitable for different data types
- Compensate for sparse labels

# Deep learning for Anomaly Detection: Theory and Applications

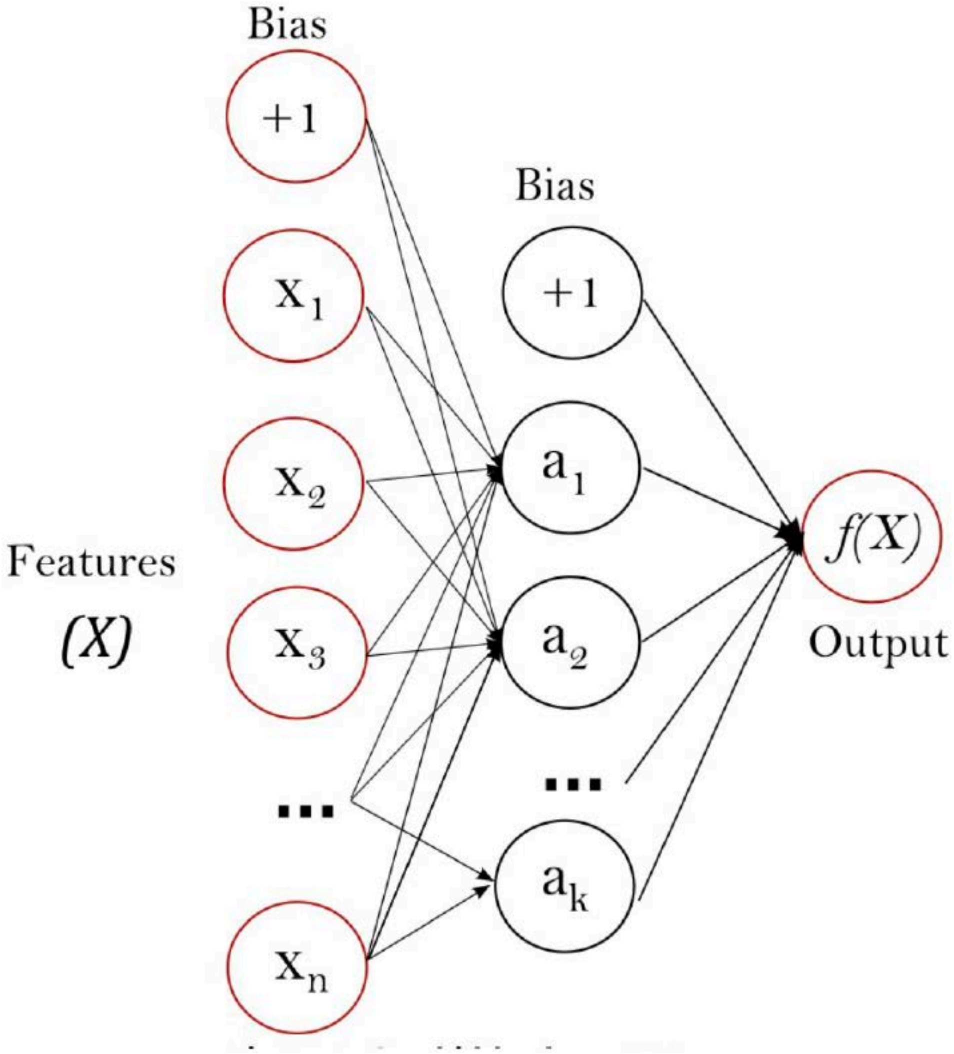
Jason J. Jung
Chung-Ang University
Seoul, Korea
j2jung@gmail.com
http://intelligent.pe.kr

# Deep learning architecture for AD

- Basic architectures
  - Multilayer perceptron (MLP)
  - Convolutional Neural Networks (CNN)
  - Recurrent Neural Networks (RNN)
- AD by deep learning architectures
  - Deep One Class (Deep OC)
  - AutoEncoder (AE)
- More ideas on AD (skipped)
  - Generative Models (VAE, GAN, Flow-based)
  - Transfer/federated learning

# MLP

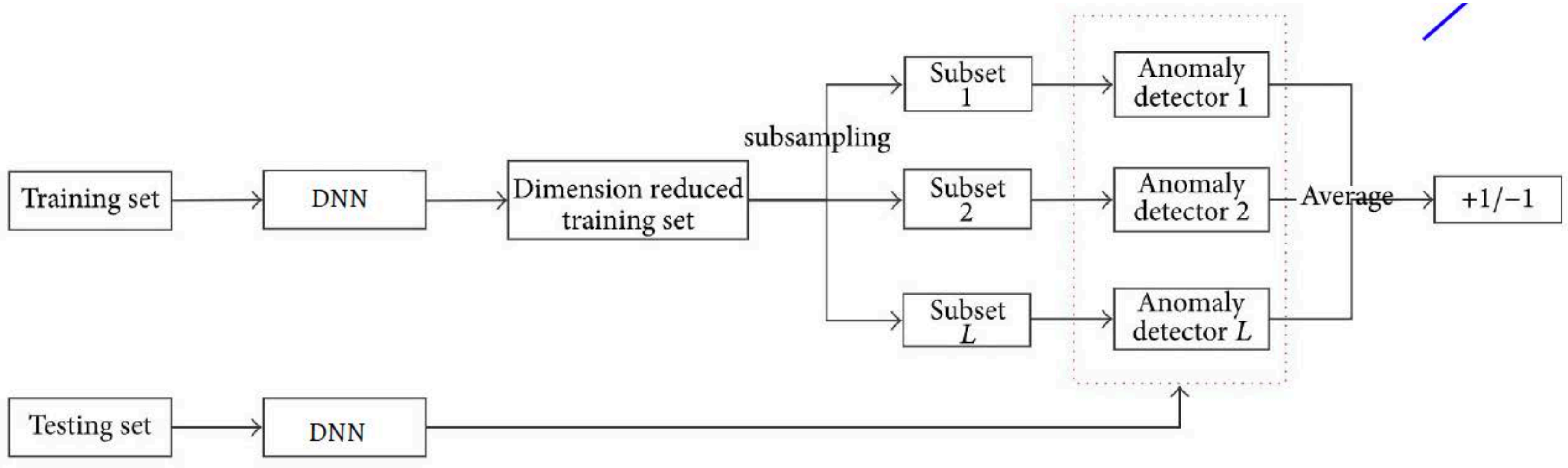
- Learn nonlinear and hierarchical discriminative features from multi-dimensional data
  - Each layer takes a linear combination of previous input and apply activation function (e.g., sigmoid, RELU, and tanh) to add nonlinearity
  - Can stack multi-layers together



# AD (k-NN) by MLP (feature extraction)

| | |
|---|---|
| **Mapping Function**<br>Map input data to a unified space | **Latent Space**<br>**f(x) is a <u>MLP</u> to map x to a low dim space** |
| **Anomaly Score**<br>Define the anomaly measure in the unified space | $d(x) = \sum_{i=1}^{Q} I\{G(f(x)) <= G(f(x_i))\} / Q$,<br>**G(·) - kth nearest neighbor distance** |
| **Decision Threshold**<br>Determine whether the data is anomalous | **x is anomaly if $d(x) > \tau$, normal o.w.** |

# AD (k-NN) by MLP (feature extraction)



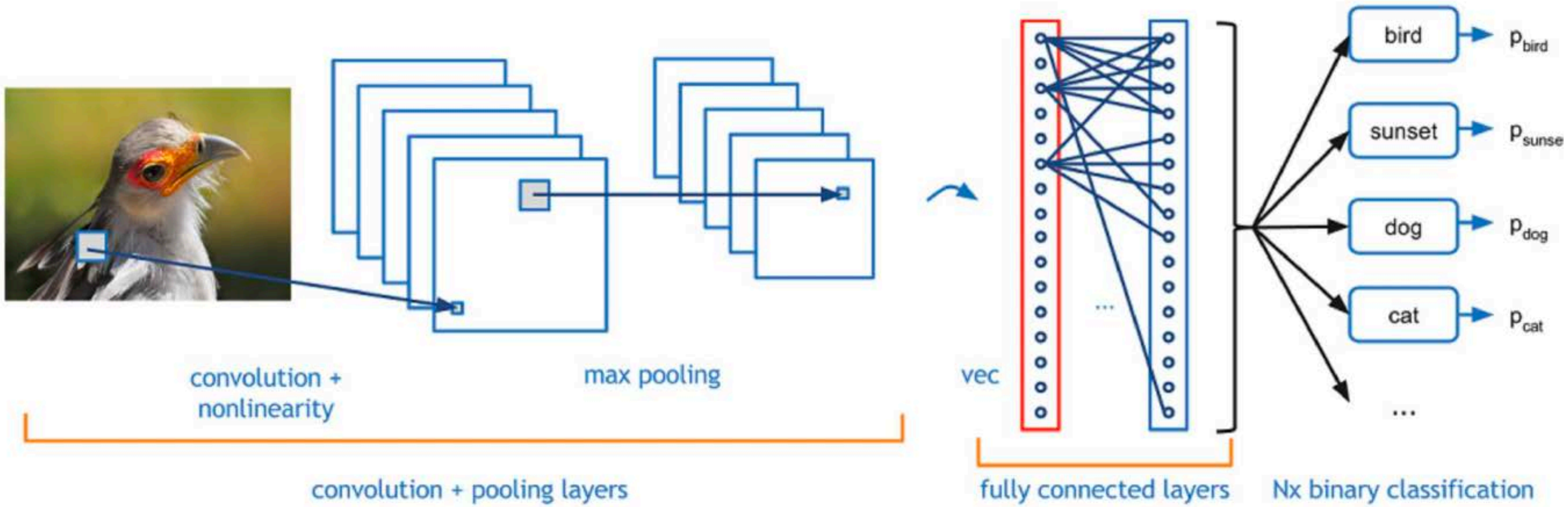
# CNN

- CNN stands for Convolutional Neural Network. It is a class of deep neural networks primarily used in the field of computer vision for tasks such as image recognition, object detection, and image classification. CNNs are designed to automatically and adaptively learn patterns and features from input data, making them particularly well-suited for tasks involving visual data. (by ChatGPT)

# CNN

- Convolutional Layers: CNNs use convolutional layers to apply convolutional operations to the input data. These operations involve sliding small filters (also called kernels) over the input data to detect features such as edges, corners, and textures.
- Pooling Layers: Pooling layers downsample the output of convolutional layers by selecting the most important information, reducing the spatial dimensions of the data while retaining essential features. Common pooling operations include max-pooling and average-pooling.
- Fully Connected Layers: After the convolutional and pooling layers, CNNs often have one or more fully connected layers, which perform high-level feature extraction and decision-making. These layers are similar to those found in traditional neural networks.
- Activation Functions: Non-linear activation functions like ReLU (Rectified Linear Unit) are commonly used in CNNs to introduce non-linearity and enable the network to learn complex patterns.
- Weight Sharing: CNNs use weight sharing, which means that the same set of filter weights is applied to different parts of the input data. This property allows CNNs to learn translation-invariant features, making them robust to variations in the position of objects within an image.

# CNN



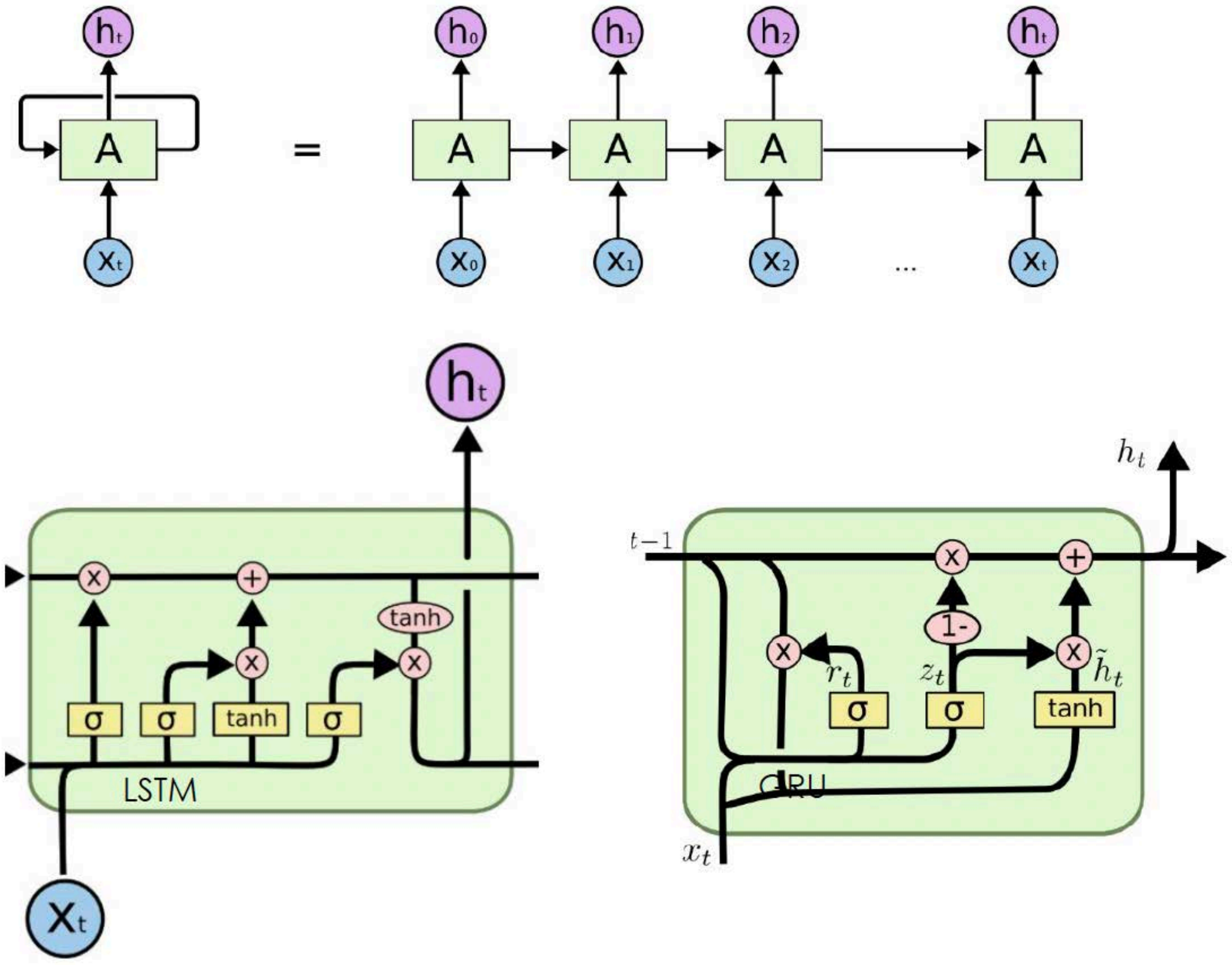
bird
sunset
dog
cat
...
$p_{bird}$
$p_{sunse}$
$p_{dog}$
$p_{cat}$
...
vec
convolution +
nonlinearity
max pooling
convolution + pooling layers
fully connected layers
Nx binary classification

# CNN

| | |
|---|---|
| **Mapping Function**<br>Map input data to a unified space | **Pre-Defined Space**<br>$F(\cdot; \mathbf{W}): x_{-a} \rightarrow x_a$, a CNN model |
| **Anomaly Score**<br>Define the anomaly measure in the unified space | **Mask area prediction error**<br>$d(x_a) = \| F(x_{-a}) - x_{-a}\|$ |
| **Decision Threshold**<br>Determine whether the data is anomalous | $x_a$ is anomaly if $d(x_a) > \tau$, normal o.w.<br>$\tau$ picked through precision/recall in validation set |

# RNN

- Learn and recognize complex context, model sequence structure with recurrent lateral connections




- LSTM
- GRU

# LSTM for Time series AD

| | |
|---|---|
| **Mapping Function**<br>Map input data to a unified space | **Pre-Defined Space**<br>$f(\cdot; W): x_{1:t-1} \rightarrow x_t$, a <u>LSTM</u> |
| **Anomaly Score**<br>Define the anomaly measure in the unified space | $d(x_t) = \| f(x_{1:t-1}) - x_t \|$ |
| **Decision Threshold**<br>Determine whether the data is anomalous | $x$ is anomaly if $(d(x) - \mu)^T \Sigma^{-1} (d(x) - \mu) > \tau$, normal o.w.<br>$\tau$ fine-tuned by precision/recall |

# Insights

Deep learning models (such as MLP, CNN and RNN) can better capture data representation, especially CNN and RNN to capture data spatial and temporal dependency.

Use as mapping functions for other deep learning models or extract features as input for traditional AD methods.

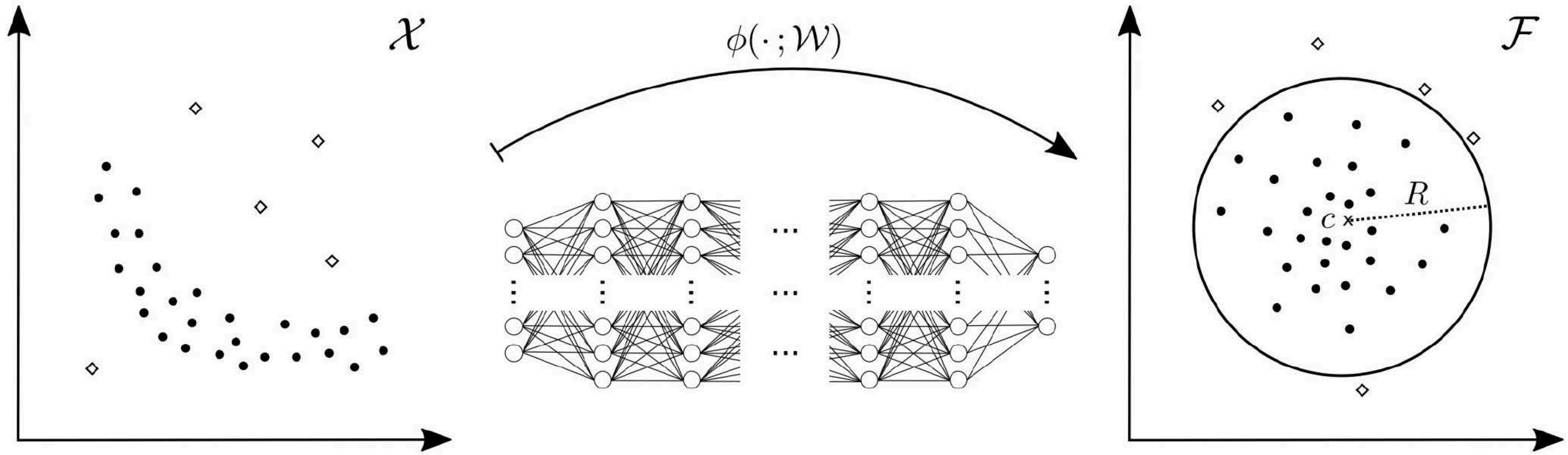
# Deep learning architecture for AD

- Basic architectures
  - Multilayer perceptron (MLP)
  - Convolutional Neural Networks (CNN)
  - Recurrent Neural Networks (RNN)
- AD by deep learning architectures
  - Deep One Class (Deep OC)
  - AutoEncoder (AE)
- More ideas on AD (skipped)
  - Generative Models (VAE, GAN, Flow-based)
  - Transfer/federated learning

# AD for DL architectures

- Deep One class classification
- Autoencoder (AE)

# Deep One Class Classification

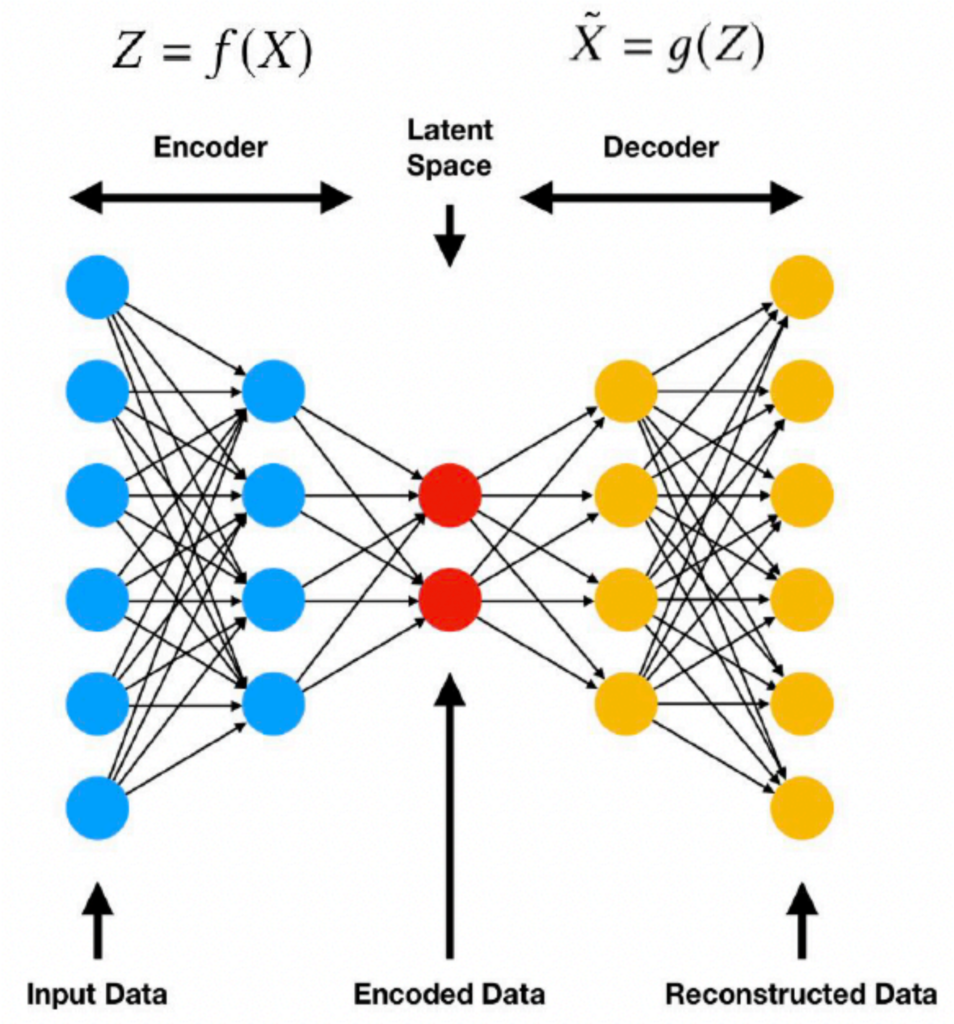


# Deep One Class Classification

| | |
|---|---|
| **Mapping Function**<br>Map input data to a unified space | **Latent Space**<br>**Neural Network $\phi(x; W)$** |
| **Anomaly Score**<br>Define the anomaly measure in the unified space | **Hypersphere**<br>**$d(x) = \phi(x) - c$** |
| **Decision Threshold**<br>Determine whether the data is anomalous | **x is anomaly if $d(x) > R^2$, normal o.w.** |

# Autoencoder (AE)

- Induce a latent representation Z to enable dimension reduction (i.e., dim(Z) < dim(X))
- Output the reconstruction of the input data
- Important to minimize the reconstruction loss, which is the differences between the original input and the reconstruction



# Autoencoder for AD

| | |
|---|---|
| **Mapping Function**<br>Map input data to a unified space | **Latent Space**<br>Encoder $f(\cdot;W_f): X \rightarrow Z$, Decoder $g(\cdot;W_g): Z \rightarrow X$ |
| **Anomaly Score**<br>Define the anomaly measure in the unified space | $d(x) = \|\| x - f(g(x)) \|\|$ |
| **Decision Threshold**<br>Determine whether the data is anomalous | x is anomaly if $d(x) > \tau$, normal o.w.<br>$\tau$ fine-tuned by precision/recall |

# Deep learning for Anomaly Detection: Theory and Applications

Jason J. Jung
Chung-Ang University
Seoul, Korea
j2jung@gmail.com
http://intelligent.pe.kr

# Outline of talk

- Basic concept on anomaly detection
- Anomaly detection on multiple time series
- Applications and experiences
  - Traffic congestion detection
  - EEG
  - Climate change
- Open issues
  - Anomaly localization
  - Early detection

# AD on multiple time series

# AD on multiple time series

- RNN-based AD for data streams
  - Assumption: These data streams are mutually independent from each other.

- In real world, most of the data streams are coupled (dependent) with each other.
- Thus, how can we detect the anomalies from multiple data streams which are dependent with each other.

# AD on multiple time series

- Basic idea
- Dependency learning among the data streams??

- Representation of the data streams as dynamic graphs over time

- Graph embedding can be applied for this issue.

# Graph embedding

- What is graphs?

# What is Graph?

- Graphs are a general language for describing and modeling complex systems

- Nodes & edges
- Node/edge types (attributes)
- Relationships
- Topology (structure)

# Examples

- Internet
- Social networks
- Information retrieval
- Biomedical/chemical graphs
- Program graphs
- Scene graphs

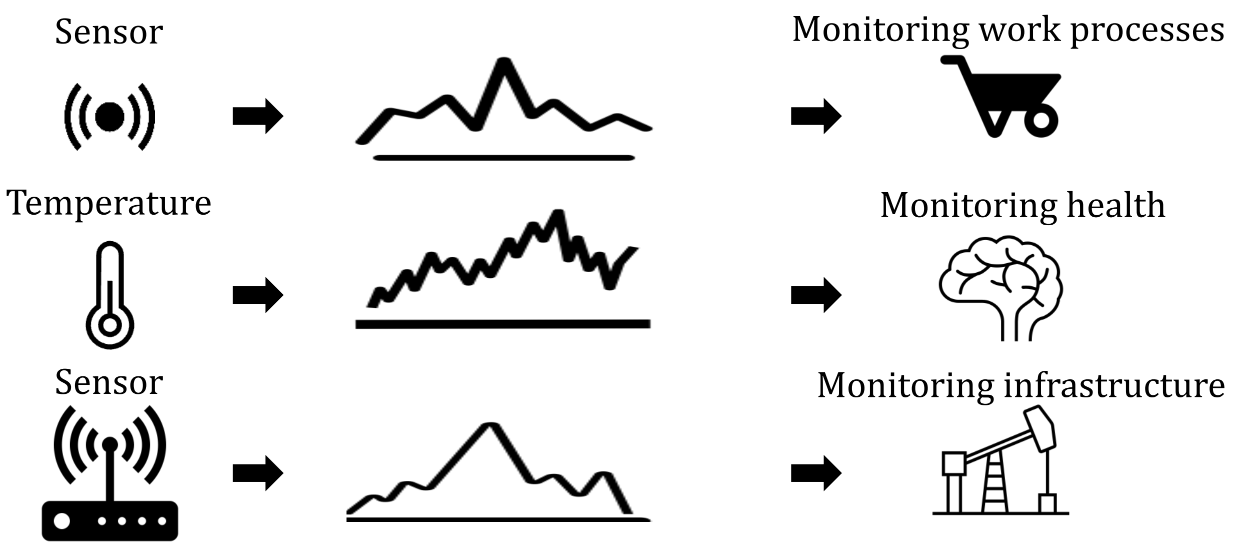
# Literature

- Gen Li, Jason J. Jung, "Deep Learning for Anomaly Detection in Multivariate Time Series: Approaches, Applications, and Challenges," Information Fusion, Vol. 91, pp. 93-102, 2023.
- Gen Li, “Graph entropy-based Early Anomaly Detection on Multiple Time series”, PhD thesis, 2022.

# 1. Introduction

## 1.1 Background

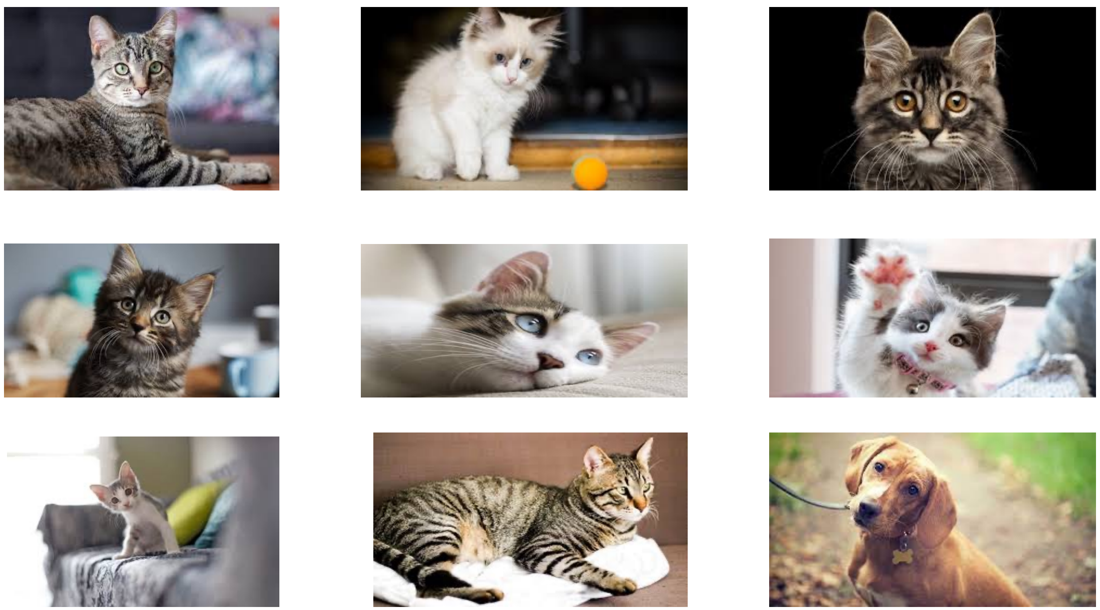
- Many application domains, ranging from finance and neurology to geology and transportation, produce large volumes of sequences of multivariate timestamped observations.



# 1. Introduction

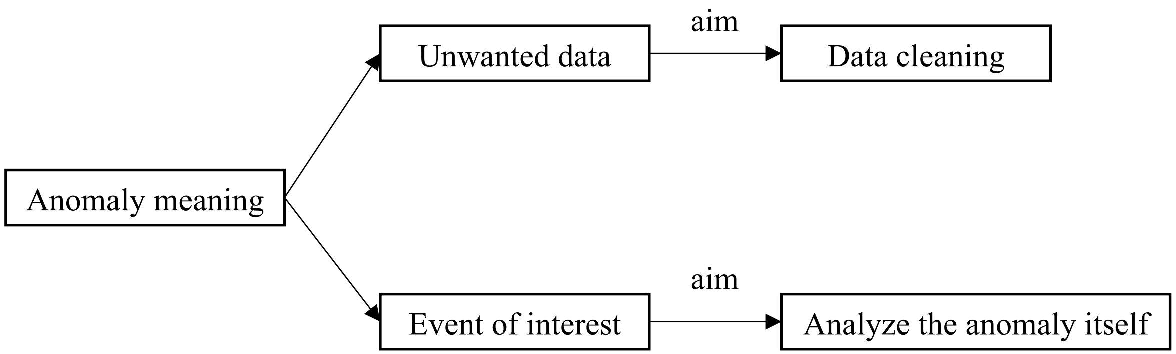
## 1.1 Background

- **Anomaly detection is to identify the data that is significantly difference with most of observations.**



## 1.1 Background

**Anomaly detection is to identify the data that is significantly different from most of observations.**



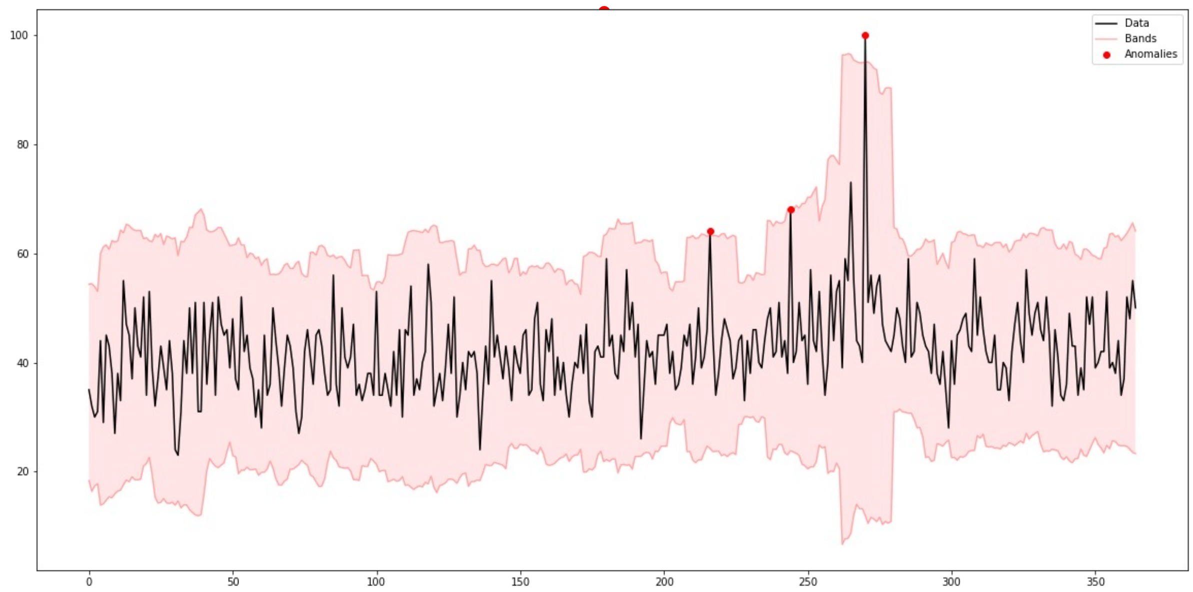

Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*.

# 1. Introduction

## 1.1 Background

**Anomaly detection is to identify the data that is significantly different from most of observations.**

### Data cleaning

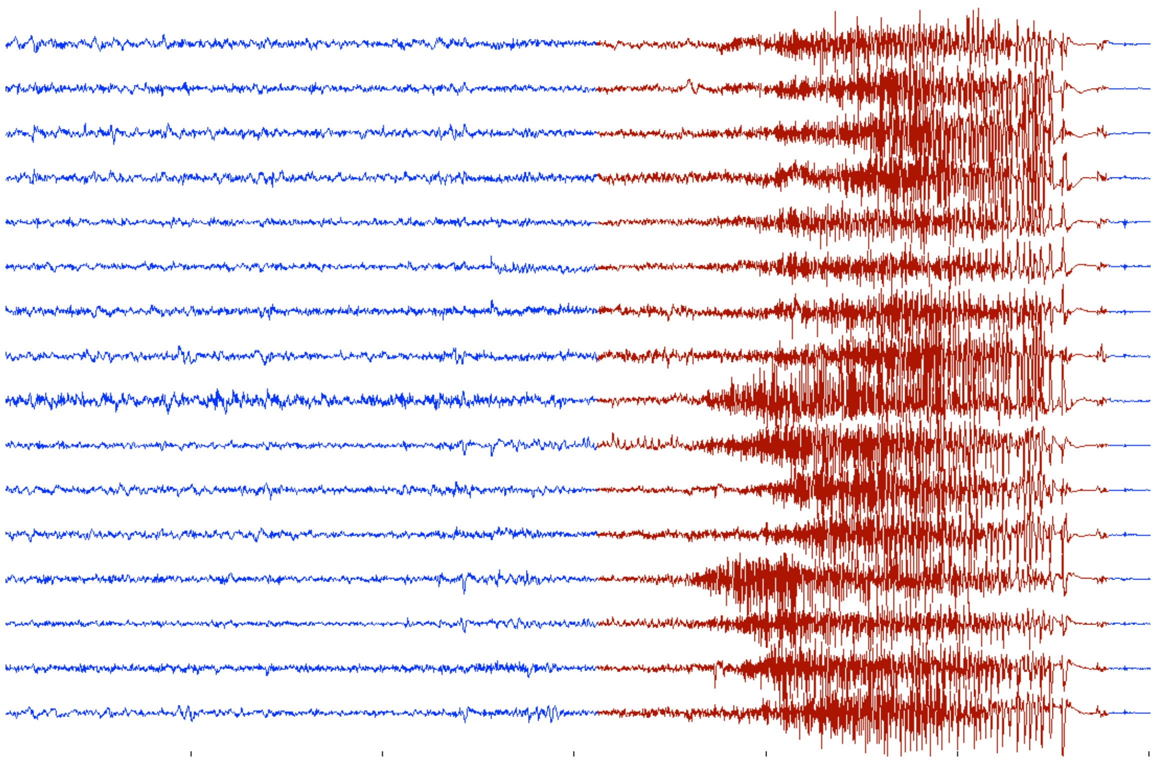


# 1. Introduction

## 1.1 Background

**Anomaly detection is to identify the data that is significantly different from most of observations.**

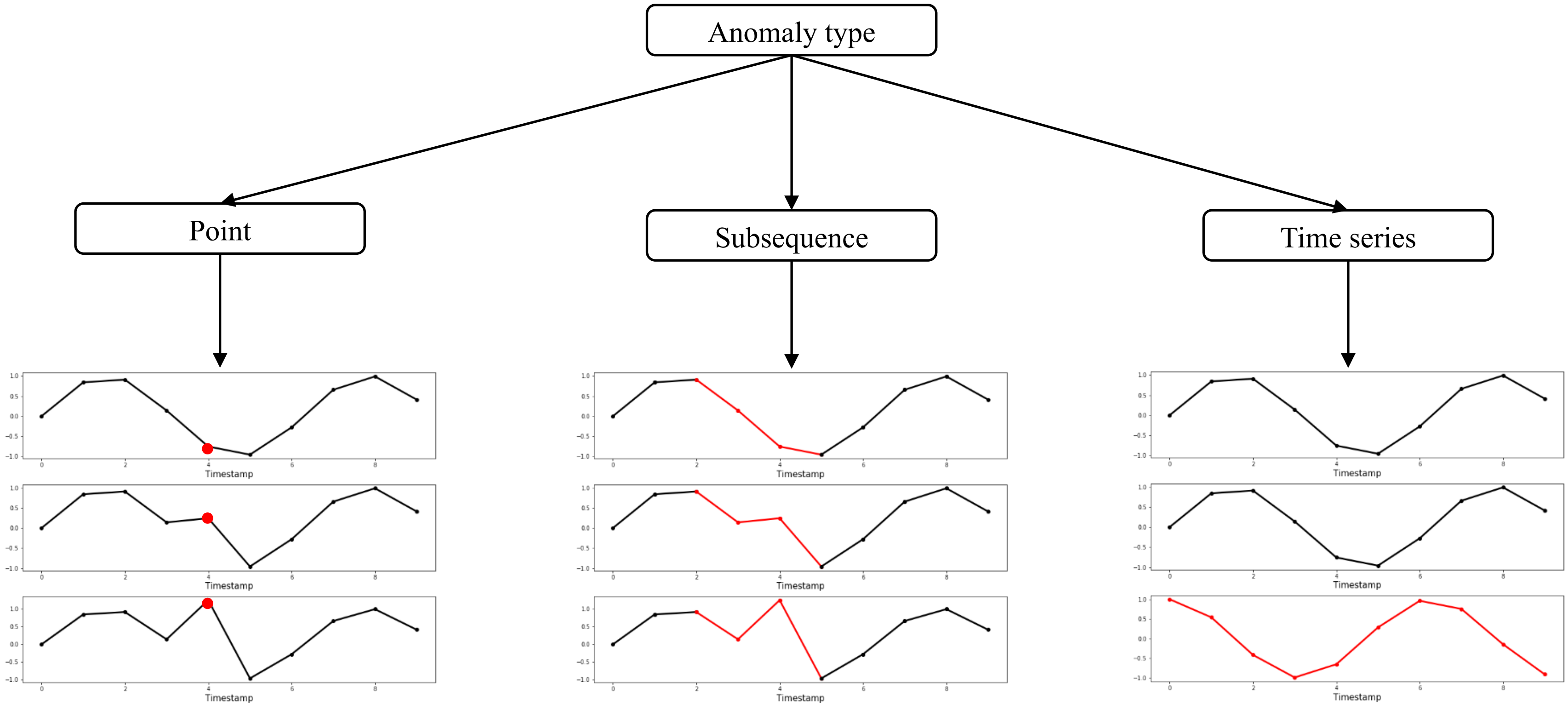
Event of interest



# 1. Introduction

## 1.1 Background

### ❖ Anomaly type

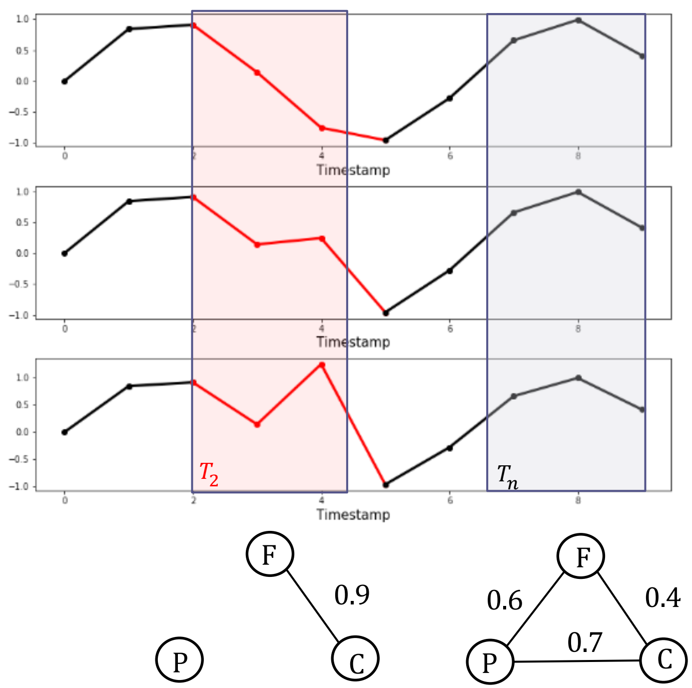



Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*.

# 1. Introduction

## 1.1 Background

Definition (Anomaly in time series): The anomaly is defined as the time interval $t_i$ where the relationships are significantly different from other time interval. It is formulated as $P(t_i) < \Theta \; or \; P(t_i) > \Theta$ in which $t_i$ is the $i^{th}$ time interval and $P(t_i)$ is the probability distribution of $t_i$, and $\Theta$ is the threshold for detecting the anomaly.

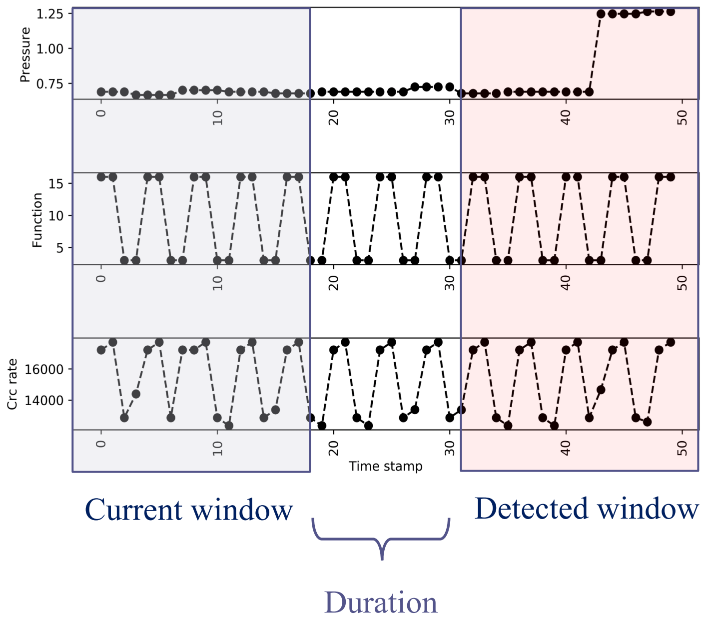



Gen Li, Jason J. Jung: *Dynamic relationship identification for abnormality detection on financial time series*, Pattern Recognition Letters, 145, pp194 – 199, 2021.

# 1. Introduction

## 1.1 Background

❖ **Detecting anomaly in an early stage.**



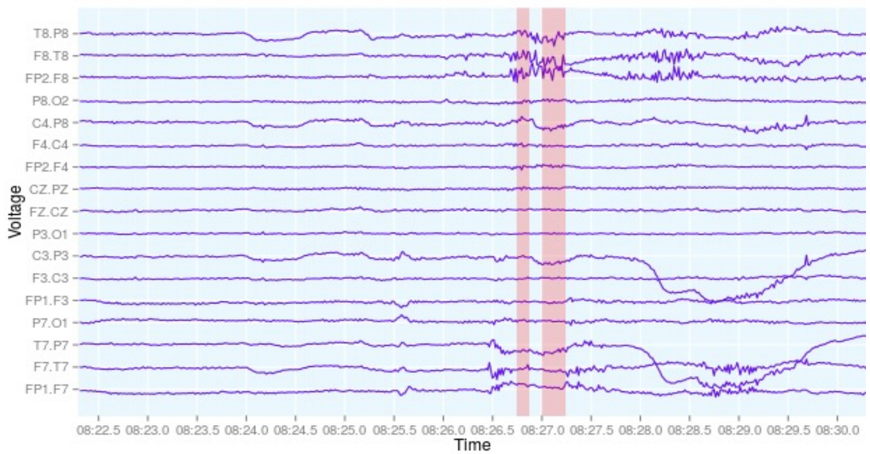

When the current window is monitored, we can detect the anomaly window before it occur.

Cerqueira, V., Torgo, L., & Soares, C. (2020). Early Anomaly Detection in Time Series: A Hierarchical Approach for Predicting Critical Health Episodes. arXiv preprint arXiv:2010.11595.

# 1. Introduction

## 1.2 Motivation

- **Anomaly detection would be much more useful when it could be done early to stop the malicious behaviors before they achieve their targets.**

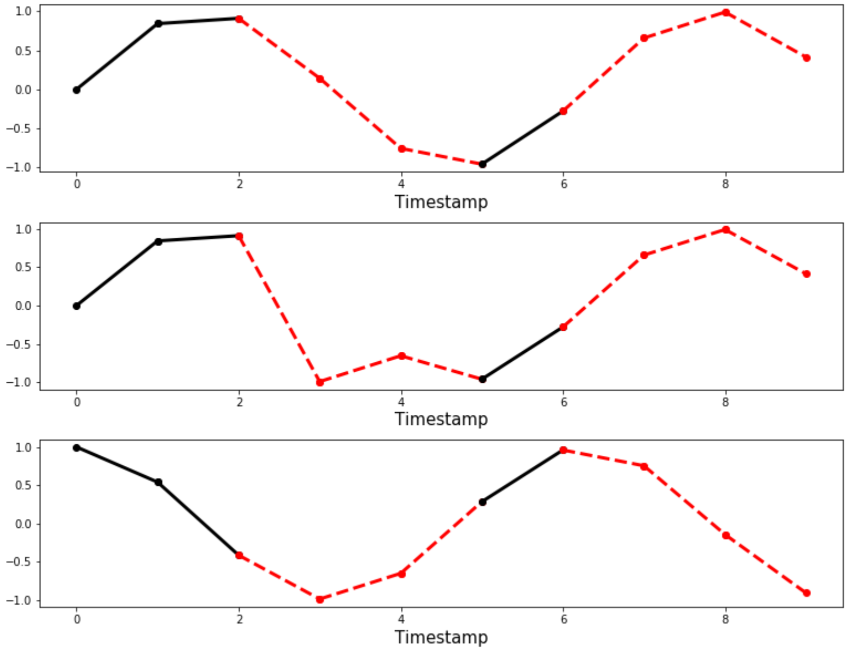



If the warning signals can be detected before the seizure, the patients can prevent the harm caused by epilepsy in advance.

# 1. Introduction

## 1.2 Motivation

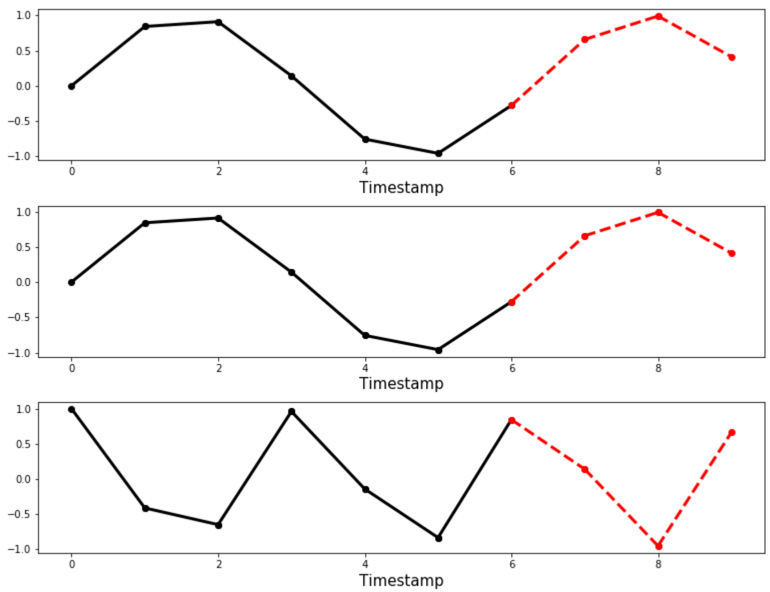
❖ **Due to the variety of abnormal patterns, existing supervised learning model is difficult to detect the anomalies that do not exhibit in the training data.**



# 1. Introduction

## 1.2 Motivation

❖ **The conventional methods for early anomaly detection is to forecast the multiple time series.**

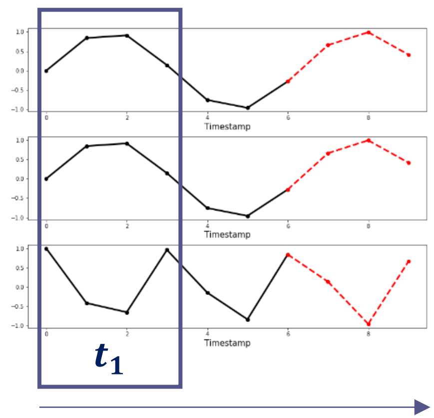


## 1.3 Research problems and contributions

- **Q1: Since the multiple time series combine the noises, the conventional patterns in time series cannot perform high performance for anomaly detection. What kinds of the abnormal patterns are extracted from the multiple time series?**
- **A1: The spurious relationships among the multiple time series are extracted as patterns.**
- **Q2: Since a correlations among the multiple time series are utilized as patterns to detect anomaly. How to model the extracted patterns for anomaly detection?**
- **A2: A dynamic graph is constructed to model these patterns.**
- **Q3: Since the conventional models for anomaly detection are based on supervised learning method and there are some limitations for the conventional models. How to construct the model for detecting the anomaly?**
- **A3:**
  - **I. The graph entropy is calculated by using the spurious relationship.**
  - **II. The graph entropy is used to measure the similarity between the graphs.**
- **Q4: Since the multiple time series is hard to predict, the conventional methods for early anomaly detection on time series perform low. How to detect the anomaly in an early stage?**
- **A4: An integrated model is proposed for early anomaly detection.**

## 2.1 Anomaly detection on time series

❖ **The conventional methods for anomaly detection on multiple time series are based on a prediction model**




**Time window**

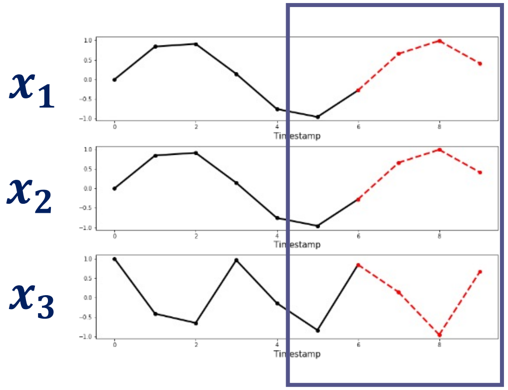
$t_{i\in[1,3]}$ → Regression model → $\hat{t_4}$ → $\left\| t_4 - \hat{t_4} \right\| > \tau$

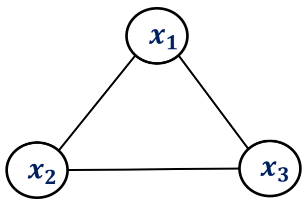
BLÁZQUEZ-GARCÍA, Ane, et al. A review on outlier/anomaly detection in time series data. *arXiv preprint arXiv: 2002.04236*, 2020.

# 2. Related work(2/2)

## 2.2 Representation of time series by dynamic graph

- **Given a graph $G = (V, E)$ where $V$ indicates the set of nodes, and $E$ indicates the set of edges.**



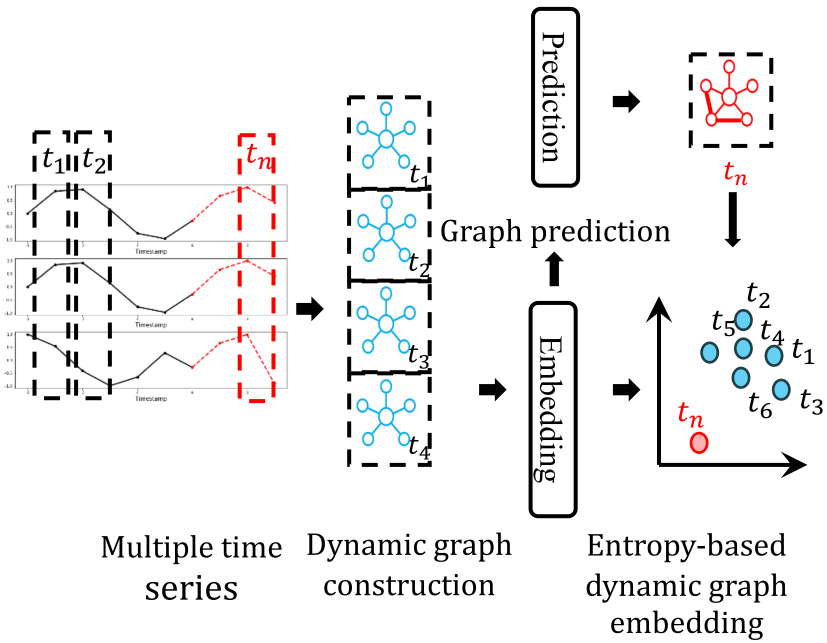




- **In this study, the nodes represent sensors and the edges represent dependency relationships. The edge from one sensor to another indicates that the first sensor is used for modelling the behavior of the second sensor.**

DENG, Ailin; HOOI, Bryan. Graph neural network-based anomaly detection in multivariate time series. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021. p. 4027-4035.

# 3. Entropy-based Early Anomaly Detection

## ❖ Architecture of the proposed method

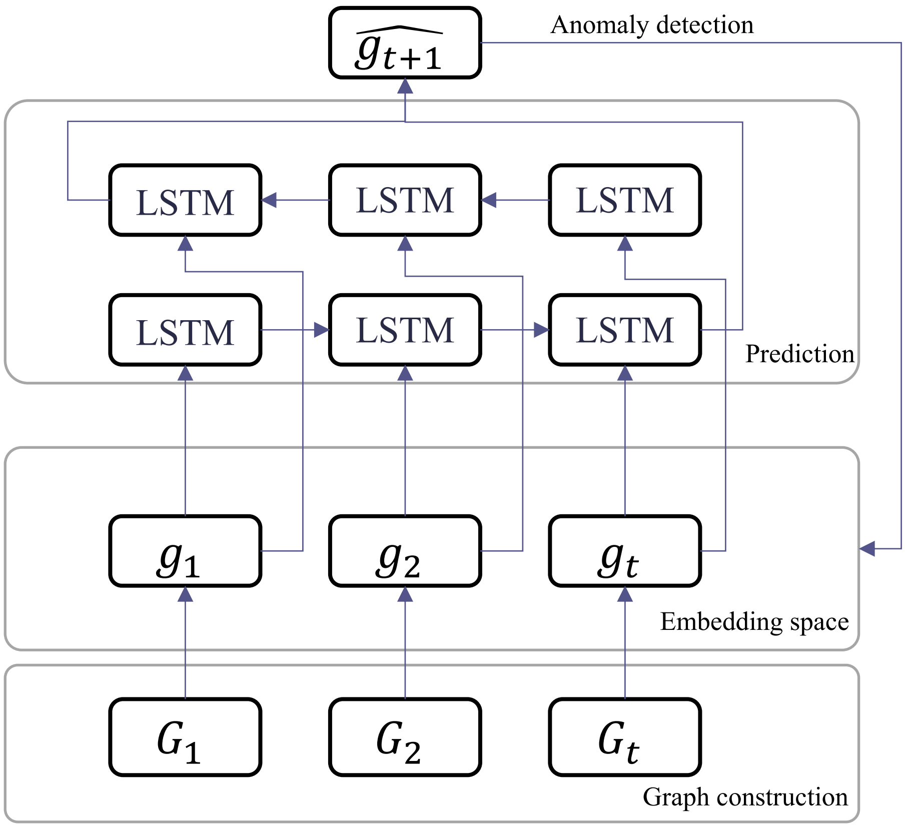



Early anomaly detection by analyzing the dynamic relationship of multiple time series.

- ❖ Dynamic graph construction
- ❖ Entropy-based dynamic graph embedding
- ❖ Bi-directional long short-term memory (Bi LSTM)-based relationship prediction
- ❖ Entropy-based anomaly detection

# 3. Entropy-based Early Anomaly Detection

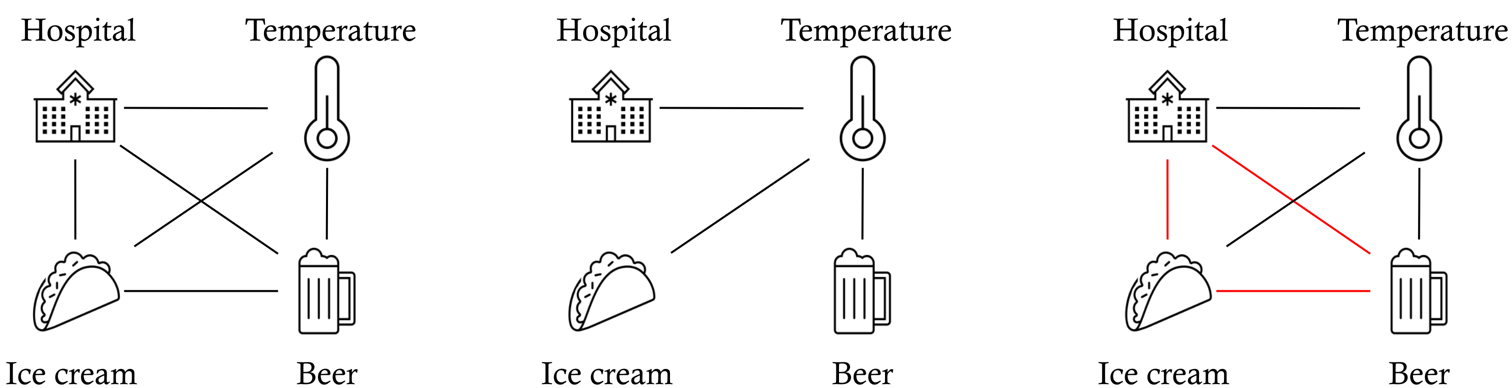
## ❖ The integrated model




- ❖ Graph construction
  - ❖ Spurious relationship
- ❖ Embedding space construction
  - ❖ Entropy-based dynamic graph embedding
- ❖ Graph prediction
  - ❖ BiLSTM model
- ❖ Anomaly detection
  - ❖ Local outlier factor

# 3. Entropy-based Early Anomaly Detection

Definition (Spurious correlation coefficient): The spurious relationship is that two time series are correlated but not causally related, which is formulated as $R(x, y) = \begin{cases} 1, & C(x, y) = 0 \\ C(x, y) - PCC(x, y), & otherwise \end{cases}$ where $C(x, y)$ indicates the causality between two time series $x$ and $y$, and $PCC(x, y)$ is Pearson correlation coefficient.




Correlation Causality Spurious relationship

Gen Li, Jason J. Jung: *Dynamic relationship identification for abnormality detection on financial time series*, Pattern Recognition Letters, 145, pp194 – 199, 2021.

# 3. Entropy-based Early Anomaly Detection

## ❖ Granger causality test

**Null hypothesis**: Two series $x$ and $y$ are not causally related.

**Regression 1**: Prediction of the current $y$ by using the historical values of $y$.

$$y_t^r = \sum_{i=1}^{t-1} \beta_i y_i + u_2$$

**Regression 2**: Prediction of the current $y$ by using the historical values of $x$ and y.

$$y_t^u = \sum_{i=1}^{t-1} \alpha_i x_i + \sum_{i=1}^{t-1} \beta_i y_i + u_1$$

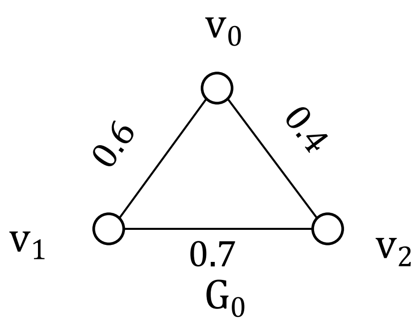
**p-value**: The probability of the null hypothesis.

**Causality discovery**: $C(x, y) = \begin{cases} 1, & p < 0.05 \\ 0, & otherwise \end{cases}$

# 3. Entropy-based Early Anomaly Detection

Definition (Vertex entropy): Given a graph $G = (V, E)$, the entropy $e(v_i)$ of a vertex $v_i$ is defined based on the weight between $v_i$ and $v_j$, which is equal to $\mathrm{e}(v_i) = \sum_{j=0}^{N} -w_{ij}\log_2(w_{ij})$.

Definition (Graph entropy): Given a graph $G = (V, E)$, the entropy $e(G)$ of a Graph $G$ is defined as the sum of the entropy of all vertices in $G$, which is equal to $e(G) = \sum_{i=0}^{N} e(v_i)$.




$$e(v_0) = -w_{1j} \sum_{j\neq \mathrm{i}, j=0}^{3} \log_2(w_{1j}) = -0.6 * \log_2 0.6 - 0.4 * \log_2 0.4 = 0.230$$

$$e(v_1) = -w_{2j} \sum_{j\neq \mathrm{i}, j=0}^{3} \log_2(w_{2j}) = -0.6 * \log_2 0.6 - 0.7 * \log_2 0.7 = 0.188$$

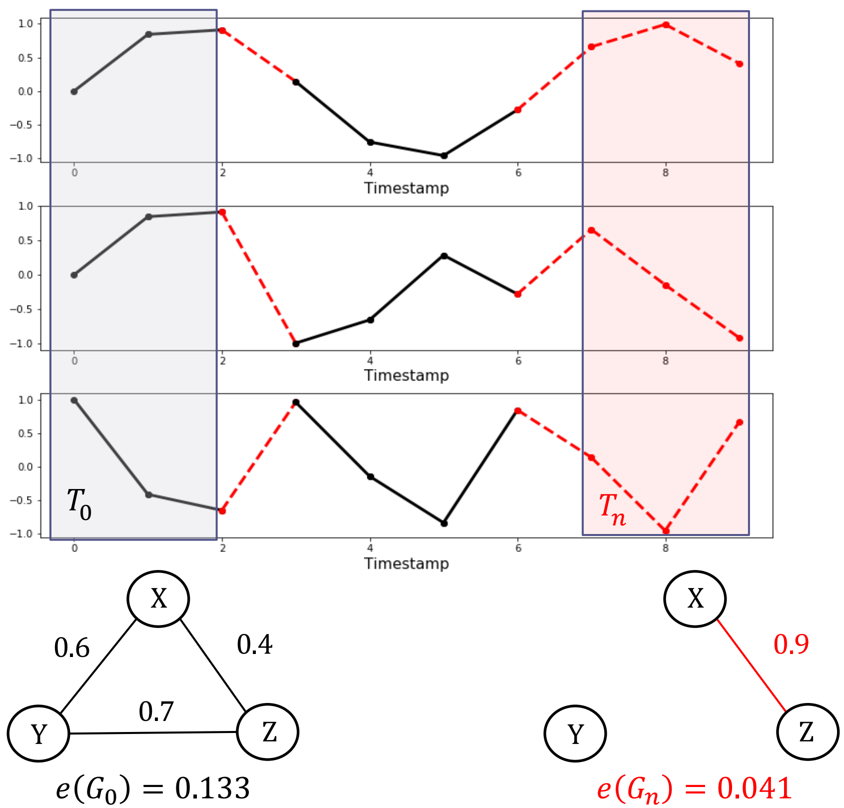
$$e(v_2) = -w_{3j} \sum_{j\neq \mathrm{i}, j=0}^{3} \log_2(w_{3j}) = -0.4 * \log_2 0.4 - 0.7 * \log_2 0.7 = 0.267$$

$$e(G) = e(v_0) + e(v_1) + e(v_2) = 0.685$$

# 3. Entropy-based Early Anomaly Detection

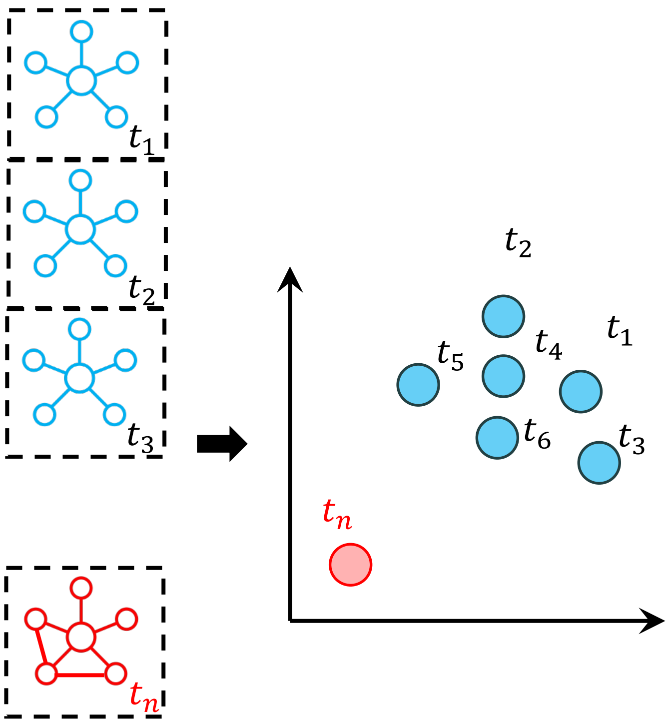
Definition (Vertex entropy): Given a graph $G = (V, E)$, the entropy $e(v_i)$ of a vertex $v_i$ is defined based on the weight between $v_i$ and $v_j$, which is equal to $\mathrm{e}(v_i) = \sum_{j=0}^{N} - w_{ij}\log_2(w_{ij})$.

Definition (Graph entropy): Given a graph $G = (V, E)$, the entropy $e(G)$ of a Graph $G$ is defined as the sum of the entropy of all vertices in $G$, which is equal to $e(G) = \sum_{i=0}^{N} e(v_i)$.



# 3. Entropy-based Early Anomaly Detection

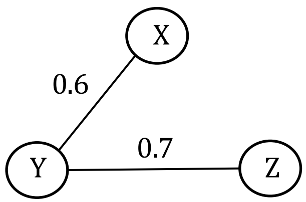
Dynamic graph embedding attempts to learn a mapping function $f: G_i \rightarrow \mathrm{g_i}$ from a dynamic graph denoted as $G = \{G_i | i \in [0, T]\}$.



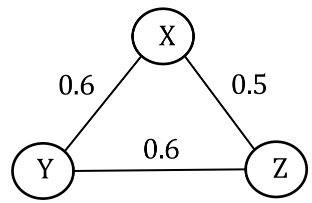

The similar graphs are close in the embedding space. The similarity between two graphs are measured by the entropy and structure.

# 3. Entropy-based Early Anomaly Detection

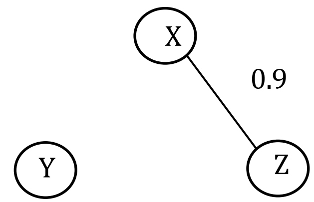
Definition (Entropy similarity): The similarity between two graphs $G_i$ and $G_j$ is defined based on their entropy, which is denoted as $d\big(e(G_i), e(G_j)\big) = \big|\big|e(G_i) - e\big(G_j\big)\big|\big|_2^2$.




$e(G_0) = 0.125$

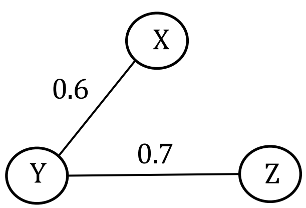



$e(G_1) = 0.139$

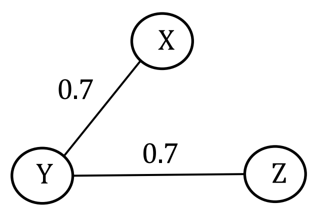



$e(G_2) = 0.041$

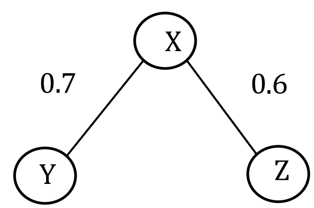
Definition (Structure similarity): The structural similarity between the graphs $G_i$and $G_j$ is defined $d\big(A_i, A_j\big) = \big|\big|A_i - A_j\big|\big|_2^2$, where $A_i$ and $A_j$ are the adjacency matrices of two graphs, respectively. If $d\big(A_i, A_j\big)$ is tiny, two graphs on the neighbor structure are comparable.




$e(G_0) = 0.125$



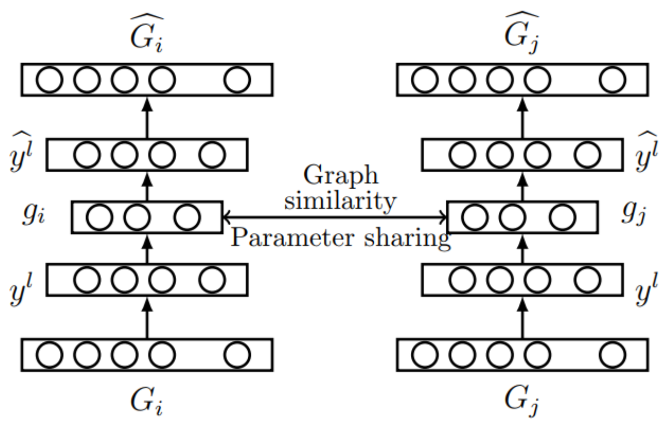

$e(G_1) = 0.108$

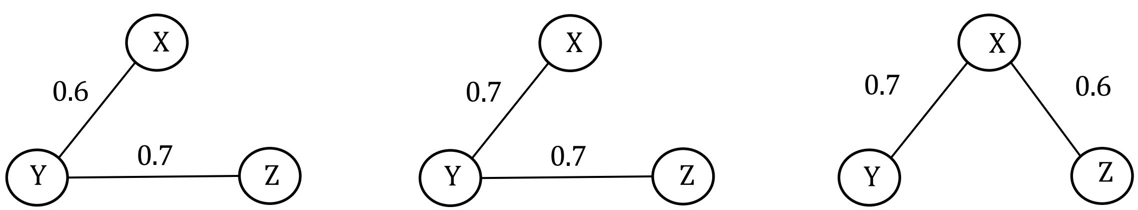



$e(G_2) = 0.125$

# 3. Entropy-based Early Anomaly Detection

Definition (Graph similarity): The similarity between two networks is calculated using the structure and entropy similarity measures, which are expressed as $d(G_i, G_j) = \left\|e(G_i) - e(G_j)\right\|_2^2 + \left\|A_i - A_j\right\|_2^2$. The goal of dynamic network learning is to minimize the distance between two graphs with comparable entropy and structure. As a result, for each graph, we determine the most comparable graph.




$e(G_0) = 0.125$ $e(G_1) = 0.108$ $e(G_2) = 0.125$

# 3. Entropy-based Early Anomaly Detection

## ❖ Model optimization

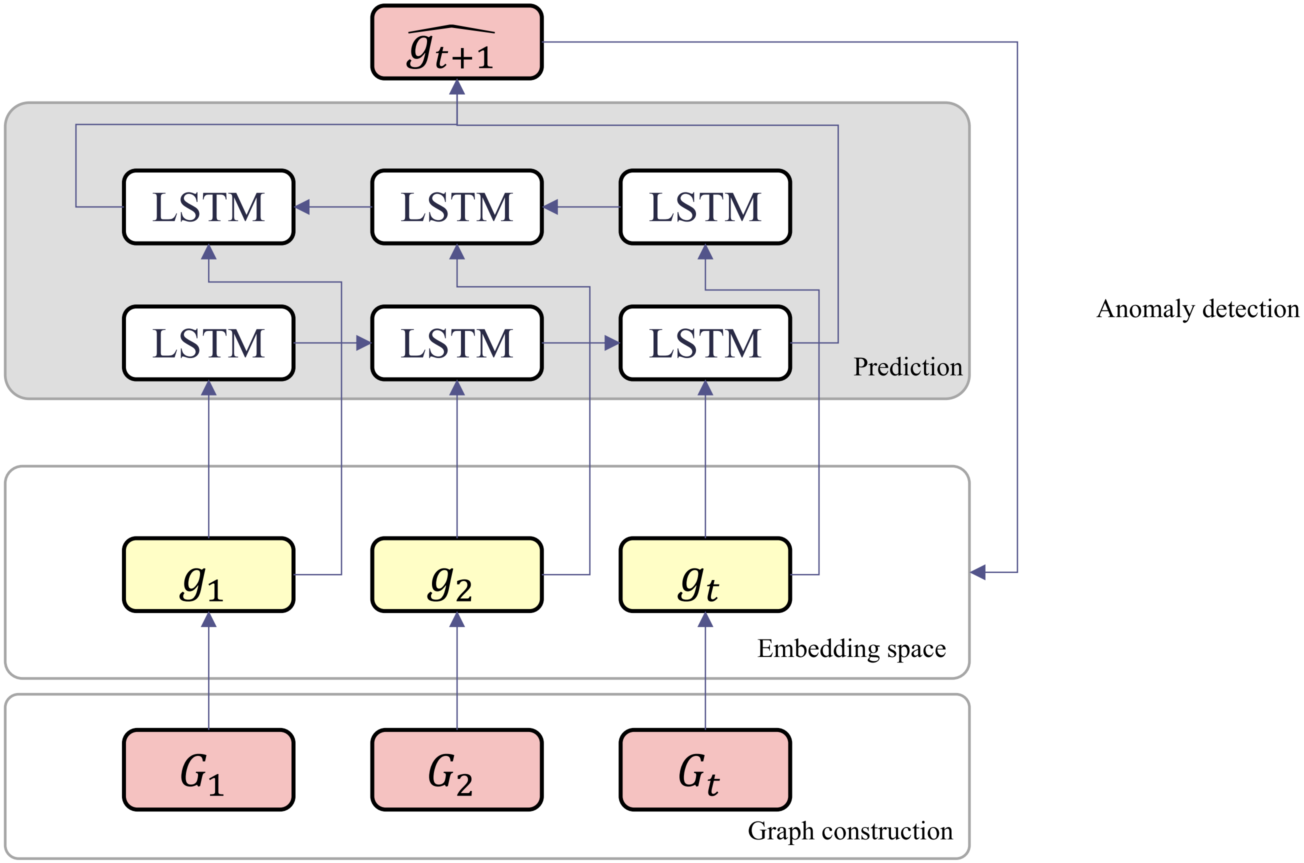
- ❖ **The optimization is carried out by using the Adam optimizer.**

$$L_1 = \frac{1}{T}\sum_{i=1}^{T}\left|\left|G_i - \widehat{G_i}\right|\right|_2^2$$

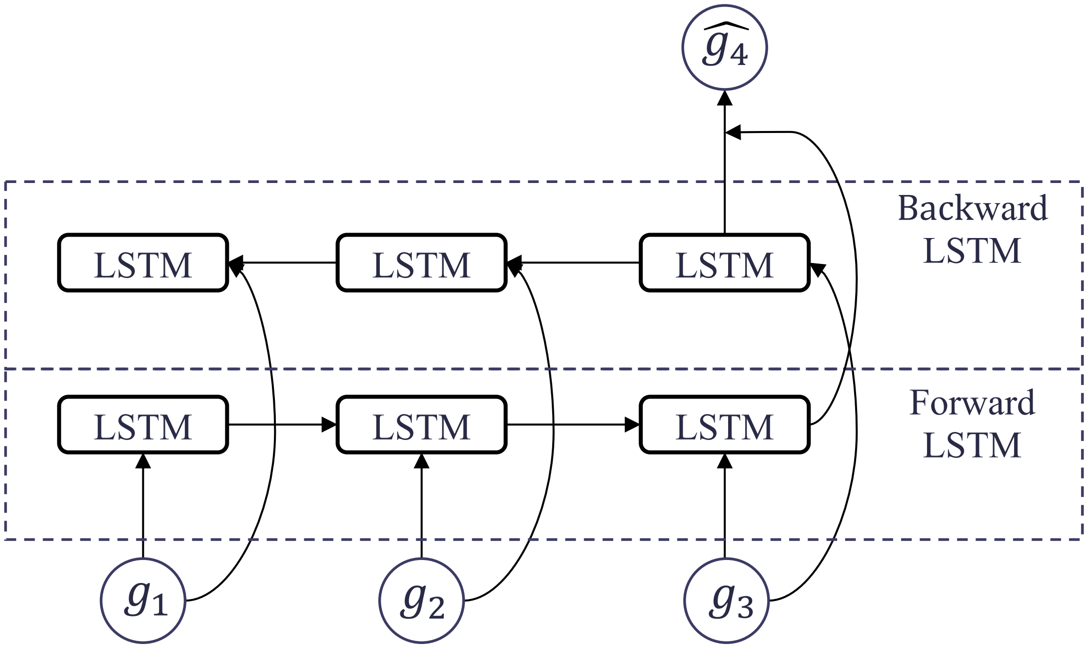
$$L_2 = \frac{1}{T}\sum_{i,j=1}^{T}\left|\left|g_i - g_j\right|\right|_2^2$$

$$L_E = \frac{1}{T}\sum_{t=1}^{T}\left|\left|G_i - \widehat{G_i}\right|\right|_2^2 + \frac{1}{T}\sum_{t=1}^{T}\left|\left|g_i - g_j\right|\right|_2^2 + \frac{1}{2}\sum_{i=0}^{I}\left(\left|\left|W^i\right|\right|_2^2 + \left|\left|\widehat{W^i}\right|\right|_2^2\right)$$

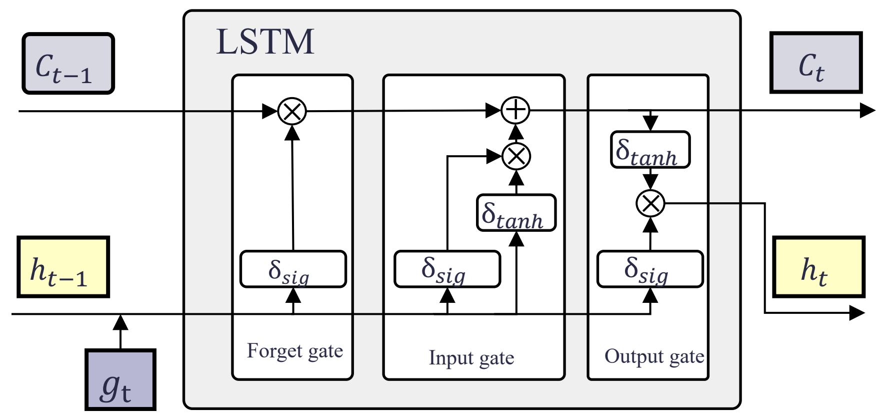
# 3. Entropy-based Early Anomaly Detection



# 3. Entropy-based Early Anomaly Detection

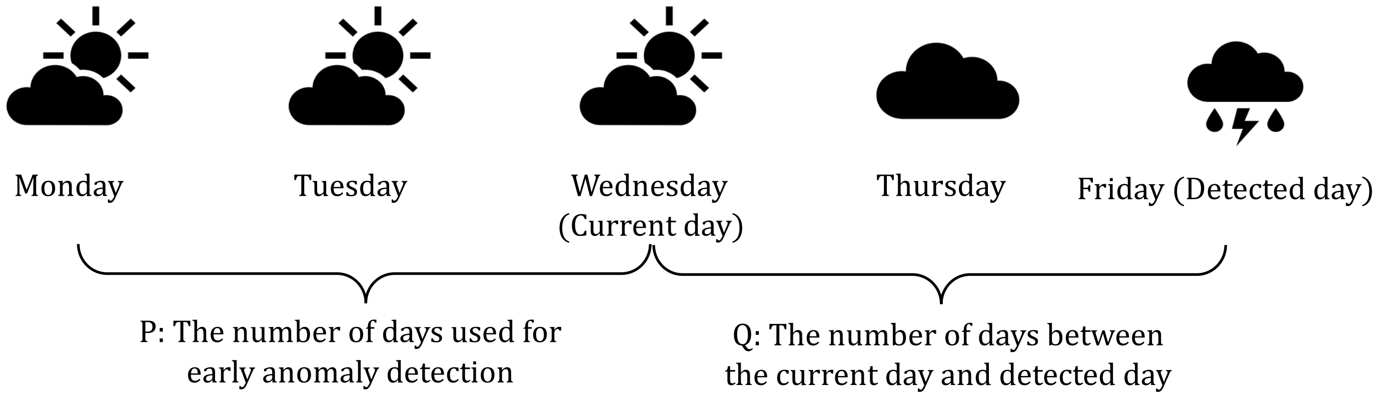



$$L_P = \frac{1}{T}\sum_{i=1}^{T} ||g_{i+n} - \widehat{g_{i+n}}||_2^2$$



# 3. Entropy-based Early Anomaly Detection

## 2.3 Dynamic relationship prediction




P: The number of days used for early anomaly detection.

Q: The number of days between the current day and detected day.

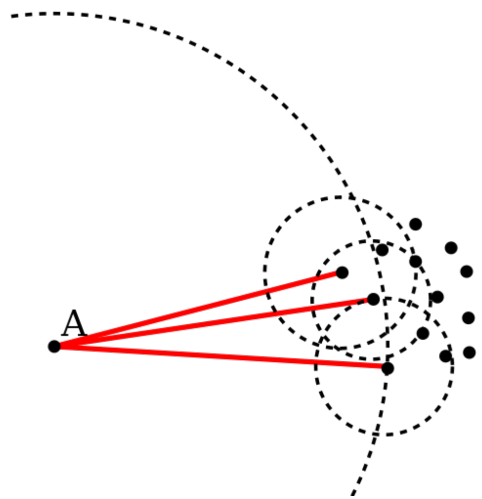
If today is Tuesday, when the P is 2 and Q is 1, it means that we used the weather of Monday and Tuesday to detect the weather of the Wednesday.

If today is Tuesday, when the P is 2 and Q is 2, it means that we used the weather of Monday and Tuesday to detect the weather of the Thursday.

# 3. Entropy-based Early Anomaly Detection

## ❖ Local outlier factors

LOF method is to detect whether the points is anomaly by comparing the density of each point and the neighborhood points.




For example, the densities of three points closest to the point $A$ are high, so that the probability of the point $A$ being detected as an anomaly is high.

Alghushairy, O., Alsini, R., Soule, T., & Ma, X. (2021). A Review of Local Outlier Factor Algorithms for Outlier Detection in Big Data Streams. Big Data and Cognitive Computing, 5(1), 1.

# 4. Experimental Results

## 4.1 Datasets

Climate datasets

The synthetic datasets

The synthetic weather datasets are generated from five different institutions, which are collected from the Australia (ACCESS), China (BCC), Canada (CanCm4), Europe (CMCC), and national meteorological research center (CNMR).

| Property | Detail |
|---|---|
| Number of features | 4 |
| Ratio of anomalies | 10% |
| Time duration | 100 years |
| Start time | 1920 |
| End time | 2020 |

## 4.1 Datasets

Climate datasets

Chinese datasets

Each dataset includes 18 features such as temperature, pressure, humidity, and so on.

| Property | Detail |
|---|---|
| Number of features | 18 |
| Number of city | 9 |
| Ratio of anomalies | 10% |
| Time duration | 30 years |
| Start time | 1990 |
| End time | 2020 |

# 4. Experimental Results

## 4.1 Datasets

Climate datasets

Korean datasets

The third datasets are Korean weather datasets provided by the Korea Meteorological Administration.

| Property | Detail |
|---|---|
| Number of features | 7 |
| Number of city | 5 |
| Ratio of anomalies | 10% |
| Time duration | 100 years |
| Start time | 1920 |
| End time | 2020 |

# 4. Experimental Results

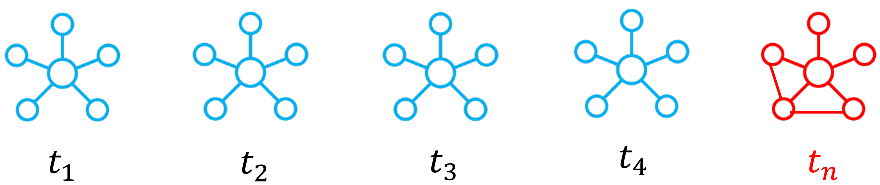
## 4.2 Baselines

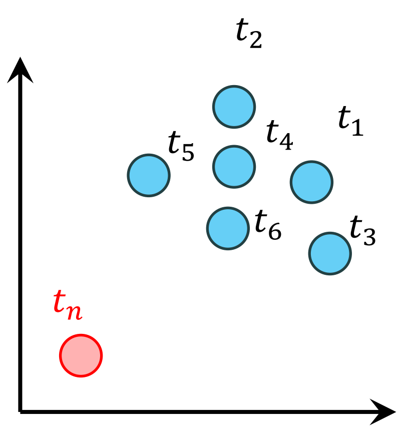
- **D2VAE – Autoencoder-based dynamic graph to vector model.**
- **D2VRNN – Recurrent neural network-based dynamic graph to vector model.**

Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., ... & Leiserson, C. (2020, April). Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 04, pp. 5363-5370).

## 4.3 Evaluation metrics

Choosing a certain number of data points as anomalies to construct a ground truth.




- ❖ Two similar graphs have a short distance.
- ❖ Normal graphs are close to each other.
- ❖ Anormal graph is far away from most graphs.

## 4.3 Evaluation metrics

After ground truth construction, the proposed model was evaluated by using the accuracy, precision, recall, and F1-score.

$$Accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n}$$

$$Precision = \frac{T_p}{T_p + F_p}$$

$$Recall = \frac{T_p}{T_p + F_n}$$

$$F1 - score = \frac{2 * Precision * Recall}{Recall + Precision}$$

- ❖ $T_p$: Number of anomalies detected to be anomalous.
- ❖ $T_n$: Number of non-anomalies detected to be non-anomalous.
- ❖ $F_p$: Number of non-anomalies detected to be anomalous.
- ❖ $F$: Number of anomalies detected to be non-anomalous.

## 4.3 Experiment conduction

The datasets are divided into training dataset and test dataset.

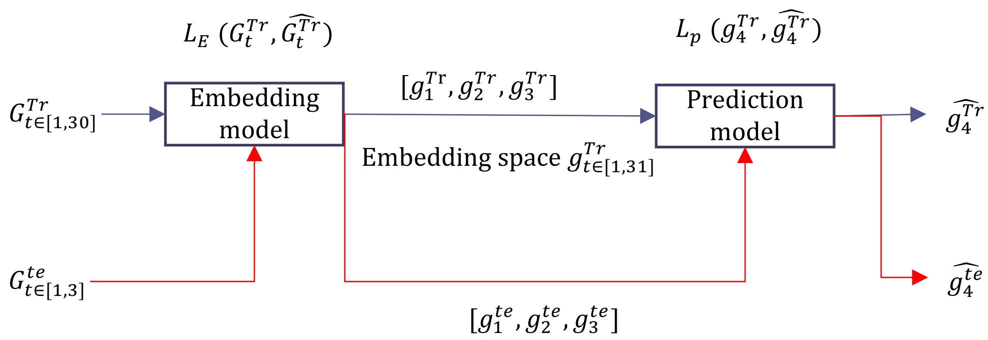
Suppose:

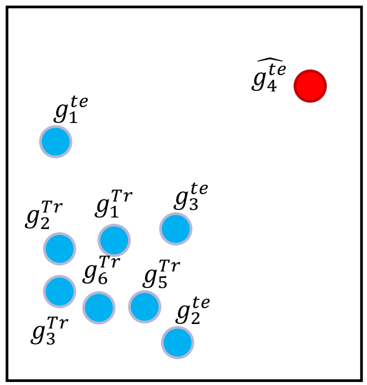
Training datasets ($G_t^{Tr}$): 1st to 30th June

Test datasets ($G_t^{Te}$): 1st to 31st July

Time window: 1 day

P and Q: 3 and 1

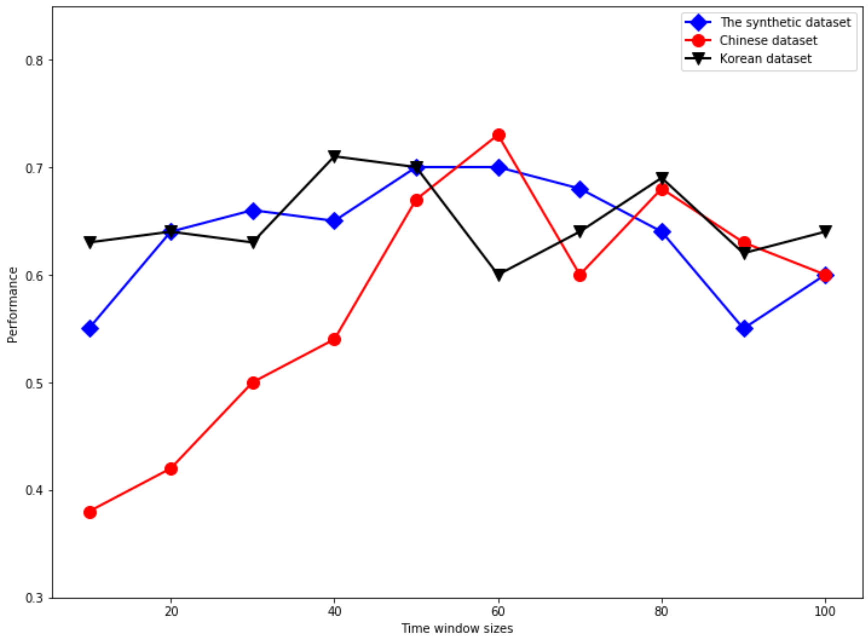


# 4. Experimental Results

## 4.4 Discussion

### ❖ Time window decision

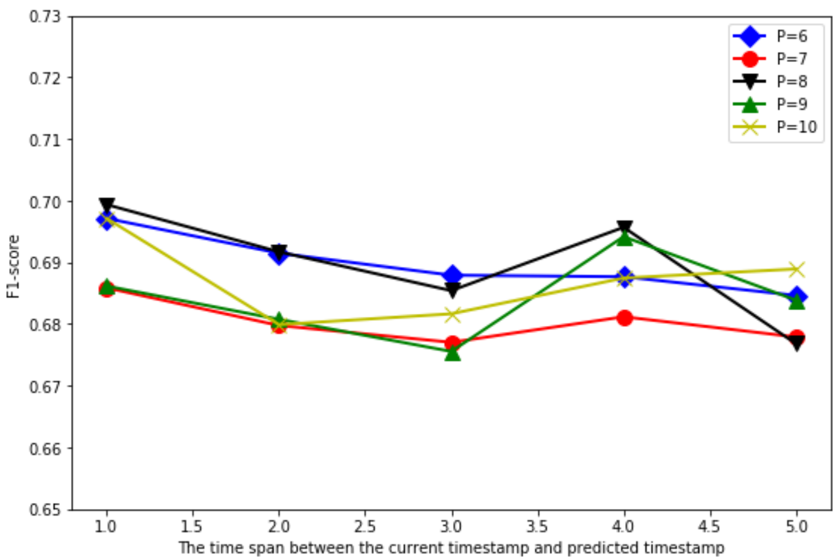



- ❖ Chinese climate dataset: 60 timestamp
- ❖ Korean climate dataset: 40 timestamp
- ❖ The synthetic dataset: 50 and 60 timestamp

# 4. Experimental Results

## 4.4 Discussion

### ❖ Experimental results on the synthetic climate datasets

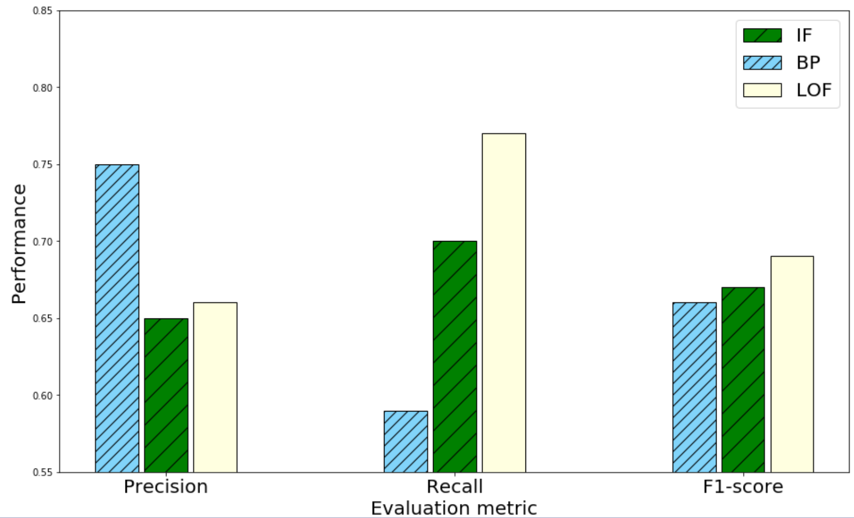



The best F1-sore is exhibited at the P of 8 and Q of 1.

# 4. Experimental Results

## 4.4 Discussion

### ❖ Experimental results on the synthetic climate datasets

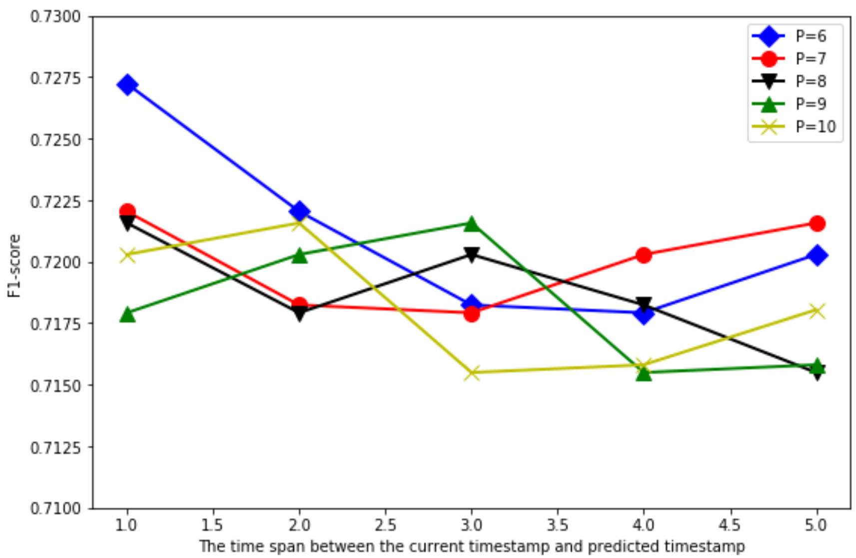



| | Proposed | Dy2AE | Dy2RNN |
|---|---|---|---|
| ACCESS | 0.66 | 0.43 | 0.43 |
| BCC | 0.71 | 0.69 | 0.62 |
| CMCC | 0.71 | 0.47 | 0.43 |
| CanCm4 | 0.68 | 0.43 | 0.46 |
| CNMR | 0.68 | 0.43 | 0.43 |

# 4. Experimental Results

## 4.4 Discussion

### ❖ Experimental results on Chinese climate datasets

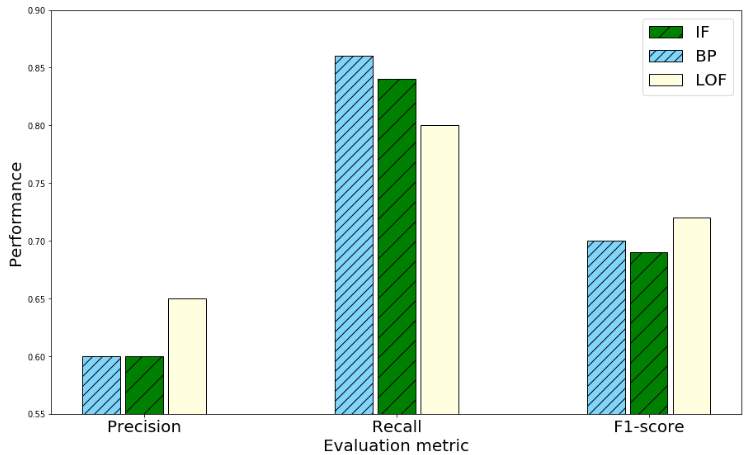



The best F1-sore is exhibited at the P of 6 and Q of 1.

# 4. Experimental Results

## 4.4 Discussion

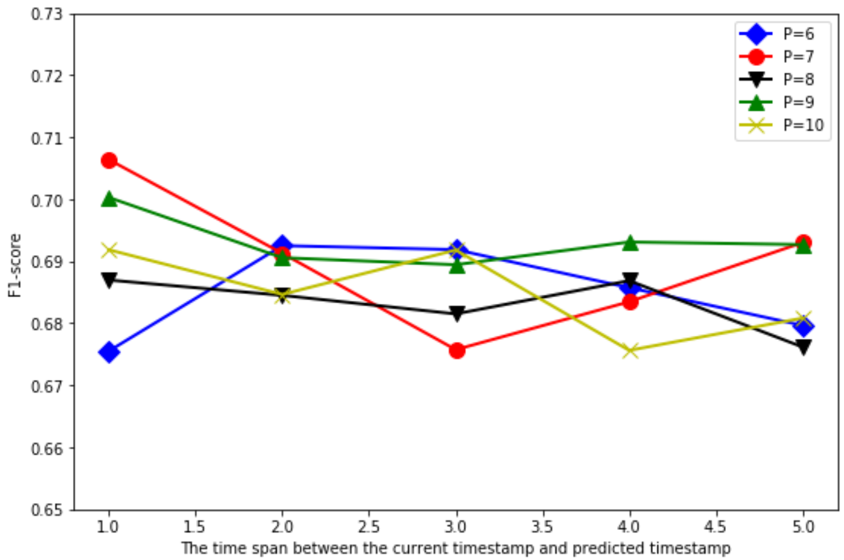
### ❖ Experimental results on the Chinese climate datasets




| | Proposed | Dy2AE | Dy2RNN |
|---|---|---|---|
| Beijing | 0.72 | 0.50 | 0.67 |
| Guangzhou | 0.68 | 0.51 | 0.63 |
| Hebei | 0.75 | 0.48 | 0.77 |
| Jiangsu | 0.74 | 0.59 | 0.72 |
| Zhejiang | 0.70 | 0.52 | 0.78 |

## 4.4 Discussion

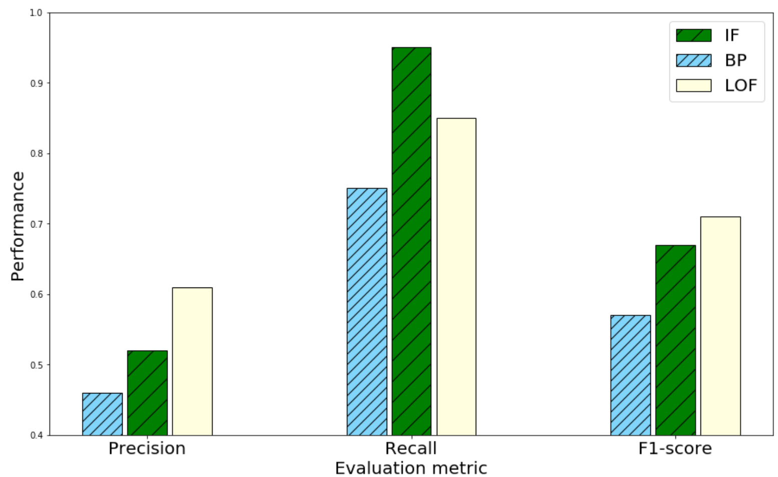
### ❖ Experimental results on Korean climate datasets




The best F1-sore is exhibited at the P of 7 and Q of 1.

## 4.4 Discussion

### ❖ Experimental results on the Korean climate datasets




| | Proposed | Dy2AE | Dy2RNN |
|---|---|---|---|
| Seoul | 0.69 | 0.45 | 0.47 |
| Incheon | 0.71 | 0.47 | 0.44 |
| Busan | 0.71 | 0.45 | 0.50 |
| Ulsan | 0.70 | 0.45 | 0.47 |
| Daegu | 0.71 | 0.44 | 0.52 |

# 5. Conclusions and limitation

❖ Conclusions

- ❖ The performance of the proposed dynamic graph embedding model outperformed the other models for early anomaly detection by 13%.
- ❖ The graph entropy measurement can improve the performance of the anomaly detection.
- ❖ Local outlier detection method outperformed the other anomaly detection methods based on proposed model by 13%.
- ❖ Dynamic relationship analysis can deal with the anomaly detection problems on the multiple time series.

# 5. Conclusions and limitation

❖ **Limitations**

- ❖ **The proposed model exhibit a low performance if the duration between the current window and detected window is long.**

- ❖ **The time window is decided by selecting several time windows to conduct the experiments.**

- ❖ **Granger causal test is to test the statistical causality between the multiple time series. Actually, this relationship is not a real causal relationship.**

- ❖ **The size of the time interval applied in this study is fixed. In the real-world environment, for different time series, selecting the appropriate length of time window is useful for relationship discovery.**

# Publication Note

- **Gen Li,** Chang Ha Lee, Jason J Jung, Young Chul Youn, David Camacho: *Deep learning for EEG data analytics: A survey*, Concurrency and Computation: Practice and Experience, 32, e5199, 2020.

- **Gen Li**, Jason J Jung: *Maximum marginal approach on EEG signal preprocessing for emotion detection*, Applied Sciences, 10, 7677, 2020.

- **Gen Li**, Jason J. Jung: *Dynamic relationship identification for abnormality detection on financial time series*, Pattern Recognition Letters, 145, pp194 – 199, 2021.

- **Gen Li,** Jason J. Jung: *Dynamic graph embedding for outlier detection on multiple meteorological time series*, Plos one, 16, e0247119, 2021.

- **Gen Li**, Jason J. Jung: *Entropy-based dynamic graph embedding for anomaly detection on multiple climate time series*, Scientific Reports, 2021.

- **Gen Li**, Jason J. Jung: *Multi-channel EEG-based seizure detection with entropy-based dynamic graph embedding,* Artificial intelligence in medicine, 2021.

- **Gen Li**, Tri-Hai Nguyen, Jason J. Jung: *Traffic incident detection based on dynamic graph embedding in vehicular edge computing*, Applied Sciences, 2021.

# Thank You

# Appendix

## Datasets

IIOT datasets

Gas pipeline datasets (ORNL), New gas pipeline datasets (NGP), Gas pipeline and water storage tank datasets (GPW), and Energy management system (EMS)

| | Property | Detail |
|---|---|---|
| ORNL | Number of features | 128 |
| | Ratio of anomalies | 10% |
| | Time duration | 5068 timestamps |
| NGP | Number of features | 12 |
| | Ratio of anomalies | 10% |
| | Time duration | 27119 timestamps |
| GPW | Number of features | 25 |
| | Ratio of anomalies | 10% |
| | Time duration | 28256 timestamps |
| EMS | Number of features | 19 |
| | Ratio of anomalies | 10% |
| | Time duration | 274626 timestamps |

## Datasets

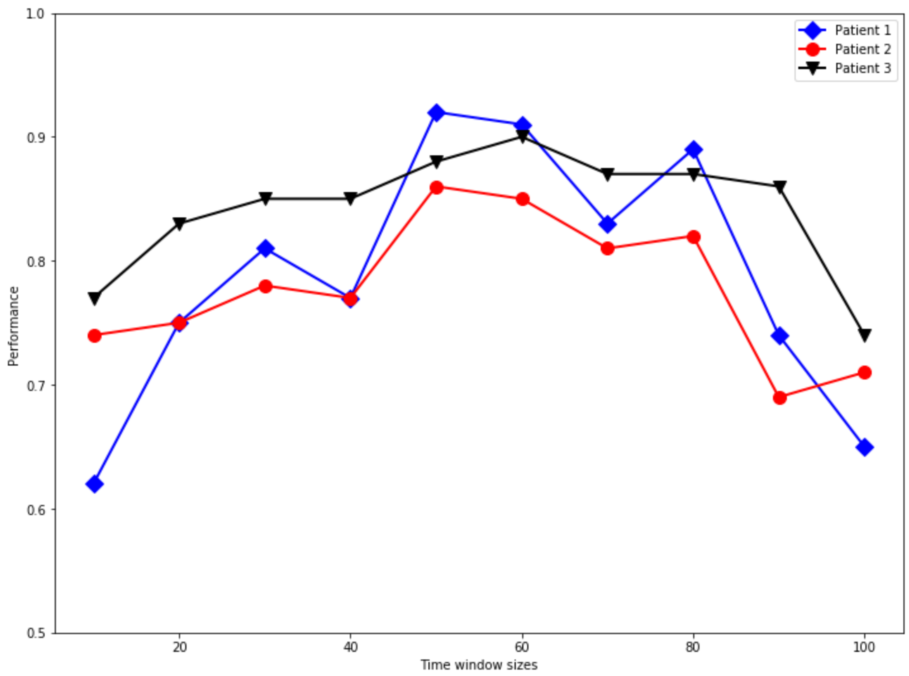
EEG

The presented seizure detection approach is applied to the CHB-MIT scalp EEG database that is available collected for research purposes.

| Property | Detail |
|---|---|
| Age | 3-22 |
| Numbers of electrode | 23-26 |
| Numbers of seizure | 146 |
| Numbers of non-seizure | 1594 |
| Number of patients | 24 |

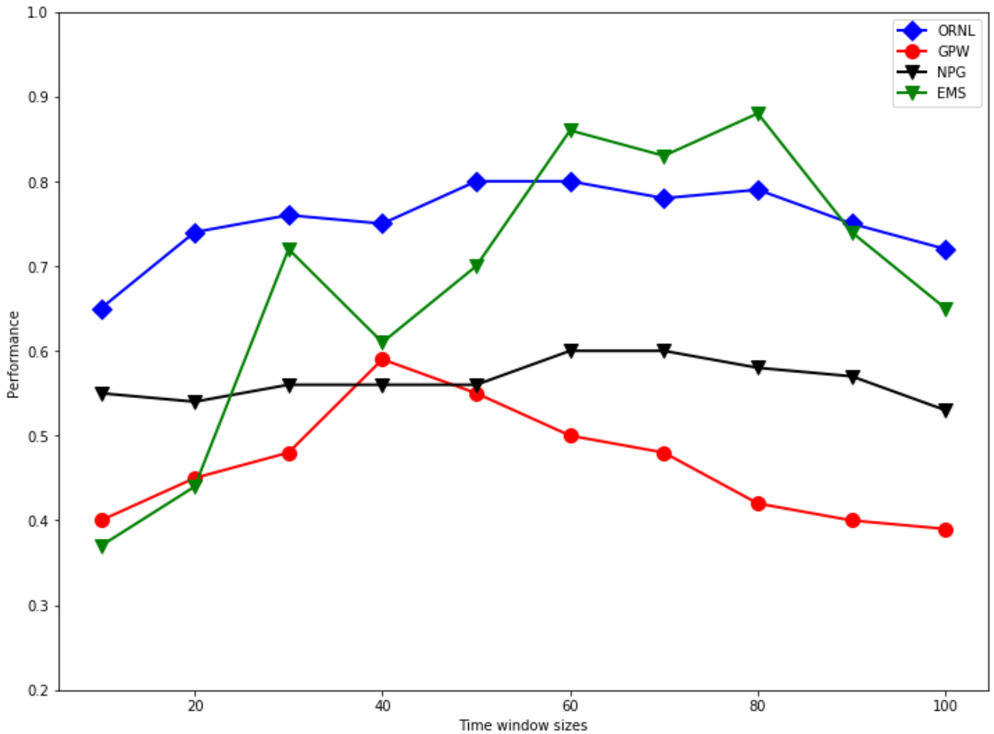
# Appendix

## ❖ Time window decision




- ❖ Patient 1: 50 timestamp
- ❖ Patient 2: 50 timestamp
- ❖ Patient 3: 60 timestamp

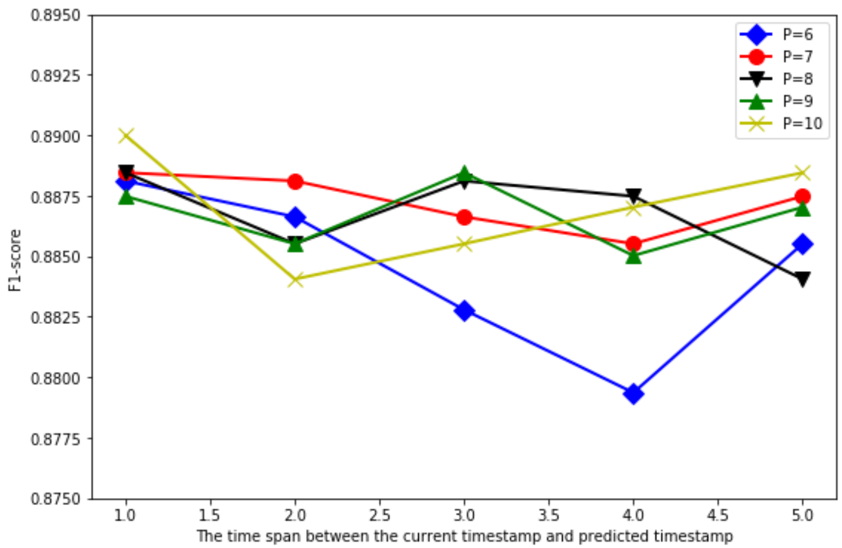
## ❖ Time window decision




- ❖ ORNL: 50 and 60 timestamp
- ❖ GPW : 40 timestamp
- ❖ NPG : 60 and 70 timestamp
- ❖ EMS : 80 timestamp

# Appendix

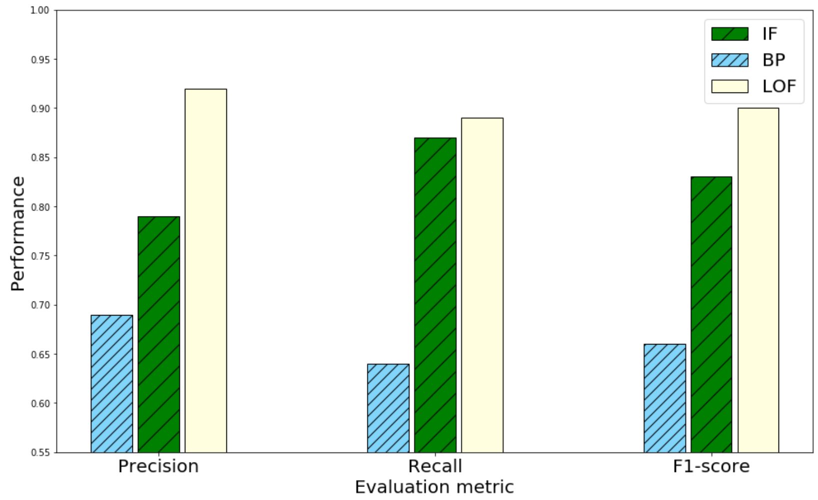
## ❖ Experimental results on EEG datasets




The best F1-sore is exhibited at the P of 10 and Q of 1.

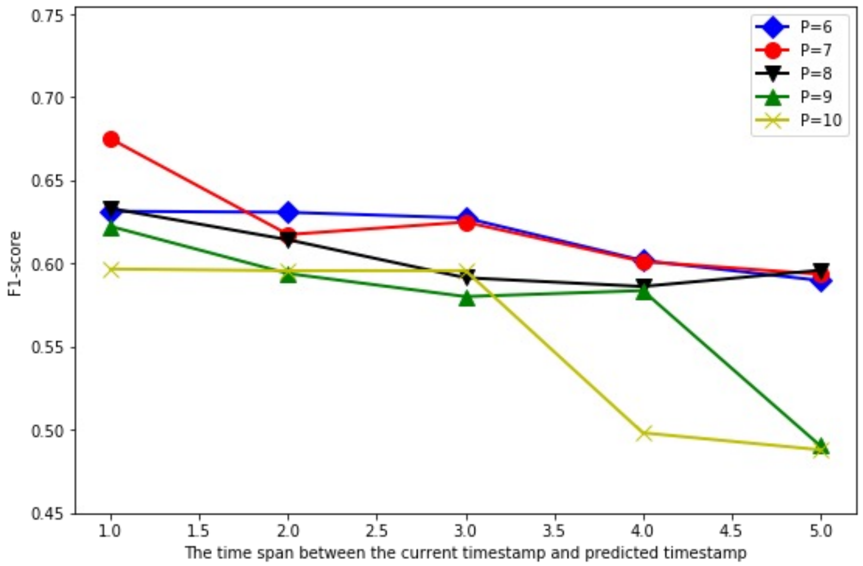
# Appendix

## ❖ Experimental results on the EEG datasets




| | Proposed | Dy2AE | Dy2RNN |
|---|---|---|---|
| Ch01 | 0.93 | 0.89 | 0.89 |
| Ch02 | 0.86 | 0.91 | 0.64 |
| Ch03 | 0.89 | 0.68 | 0.89 |
| Ch04 | 0.93 | 0.87 | 0.90 |
| Ch05 | 0.91 | 0.89 | 0.67 |

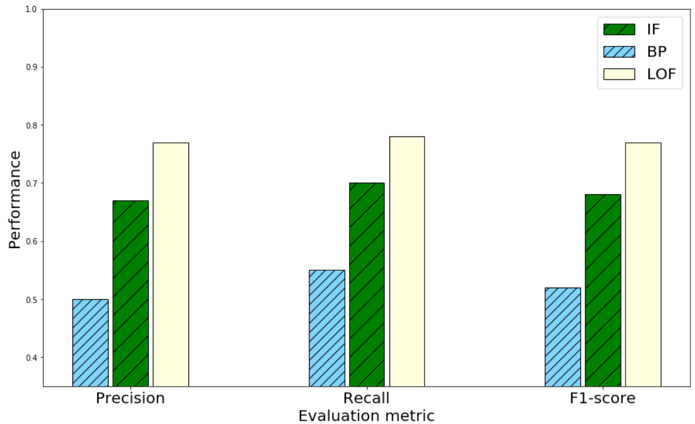
## ❖ Experimental results on IIOT datasets



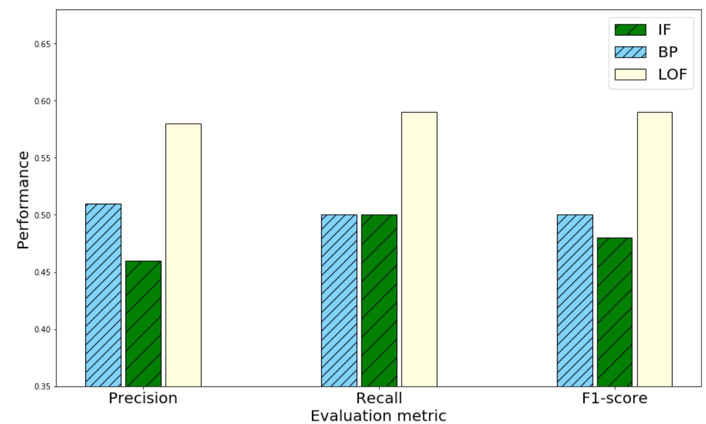

The best F1-sore is exhibited at the P of 7 and Q of 1.

# Appendix

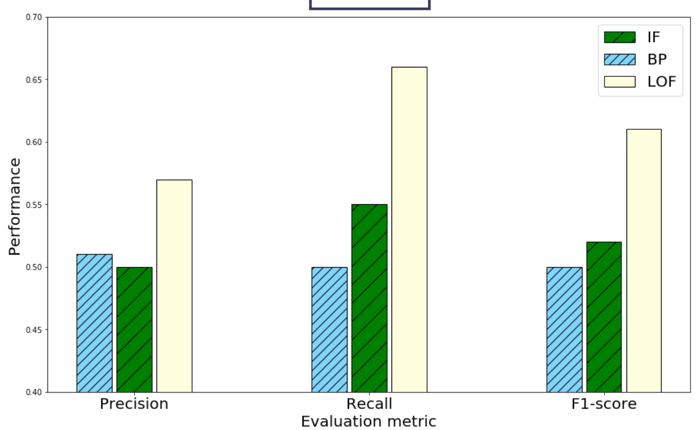
## ❖ Experimental results on the IIOT datasets



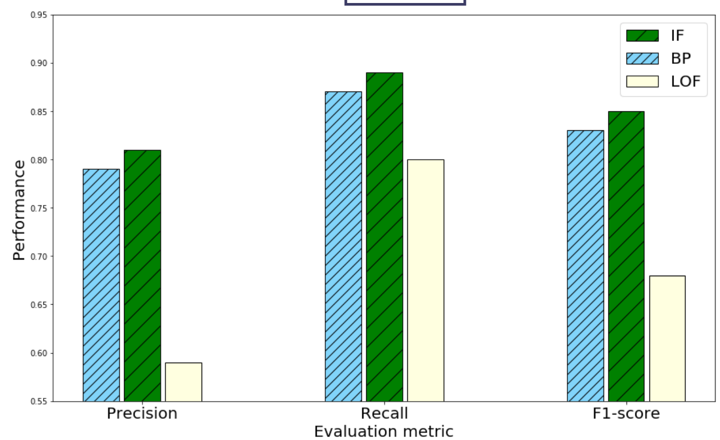

ORNL




GPW




NPG




EMS

# Appendix

❖ **Experimental results on IIOT climate datasets**

| | **Proposed** | **Dy2AE** | **Dy2RNN** |
|---|---|---|---|
| ORNL | 0.80 | 0.41 | 0.56 |
| GPW | 0.57 | 0.62 | 0.51 |
| NPG | 0.66 | 0.68 | 0.67 |
| EMS | 0.68 | 0.65 | 0.87 |