

Anomaly Detection for Time Series with Autoencoder

2022. 03. 25

발표자: 조 경 선



발표자 소개



❖ 이름: 조경선

- 고려대학교 산업경영공학과 석사과정(2020.09~)
- 데이터마이닝 및 품질애널리틱스 연구실(김성범 교수님)

❖ 관심연구분야

- Named Entity Recognition
- Anomaly Detection for Time Series

❖ 연락처

- abc0323@korea.ac.kr

목차

I. Introduction

- Anomaly Detection for Time Series

II. Autoencoder

III. Anomaly Detection for Time Series with Autoencoder

- LSTM-Autoencoder
- Autoencoder Ensembles
- Autoencoder within An Adversarial Training Framework

IV. Conclusion

목차

I. Introduction

- Anomaly Detection for Time Series

II. Autoencoder

III. Anomaly Detection for Time Series with Autoencoder

- LSTM-Autoencoder
- Autoencoder Ensembles
- Autoencoder within An Adversarial Training Framework

IV. Conclusion

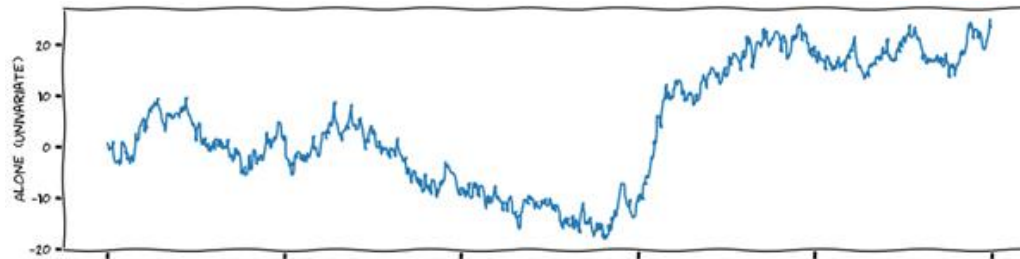
Introduction

Time Series

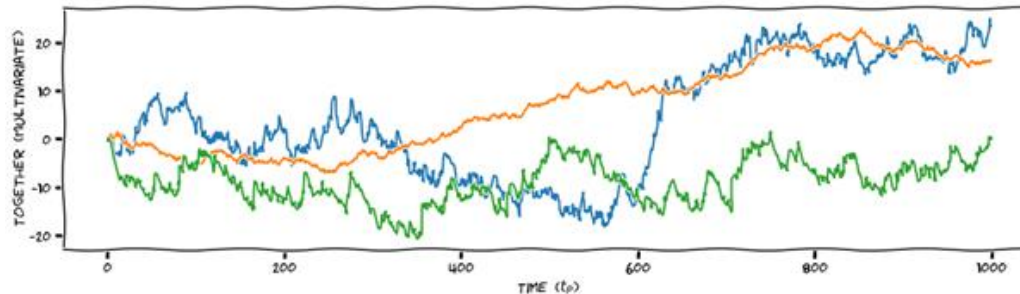
❖ 시계열(Time series) 데이터란?

- 관측치가 **시간적 순서**를 가진 데이터 셋의 집합
 - ✓ 단변량 시계열(Univariate time series): 동일한 간격의 시간의 증가에 대해 순차적으로 기록된 한 개의 변수 관측치로 구성된 시계열
 - ✓ 다변량 시계열(Multivariate time series): 동일한 간격의 시간의 증가에 대해 순차적으로 기록된 두 개 이상의 변수 관측치로 구성된 시계열
- 센서, Edge devices 등 산업 전반의 다양한 도메인에서 발생하는 데이터
- 시계열 데이터를 발생시키는 어플리케이션의 증가로 **시계열 데이터의 이상 탐지(Anomaly Detection)**에 대한 수요 증가

Univariate time series



Multivariate time series

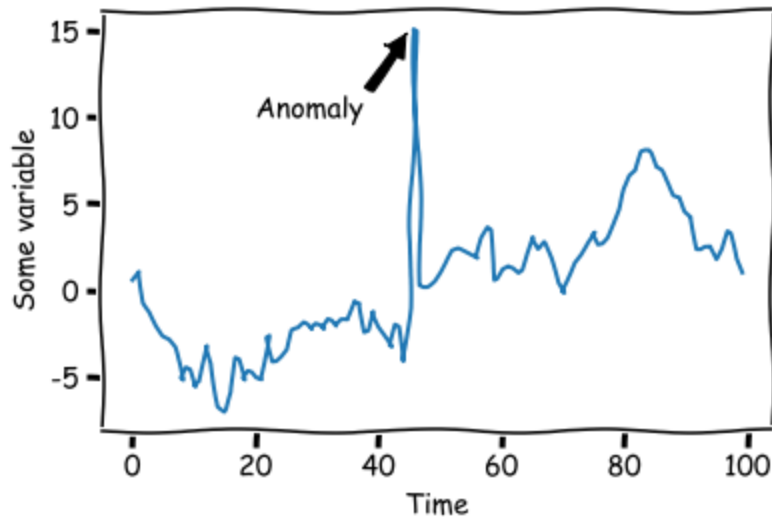


Introduction

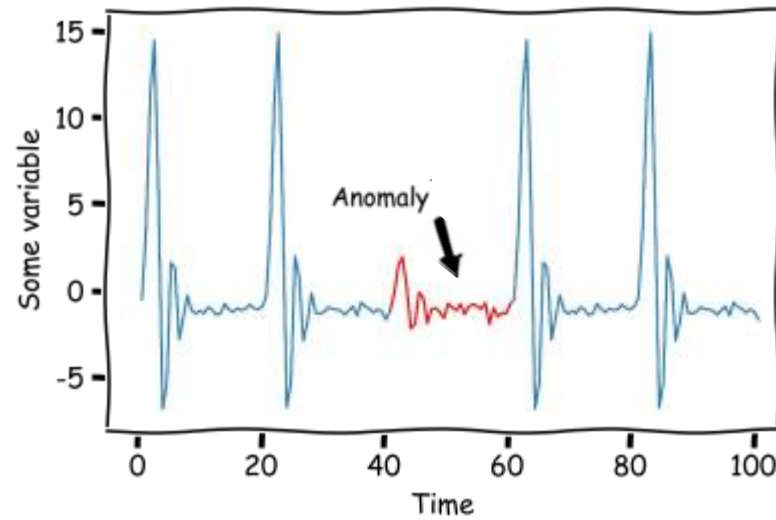
Anomaly Detection for Time Series

❖ 시계열 데이터의 이상 현상 정의

- 시스템이 비정상적으로 작동하는 **시점(point)** 또는 **기간(duration)**으로 정의
 1. 시점 이상: 비정상적인 값에 도달한 단일 시점 또는 연속적인 시점의 집합
 2. 기간 이상: 특정 구간에서 이상 현상 발견



<Point Anomaly>



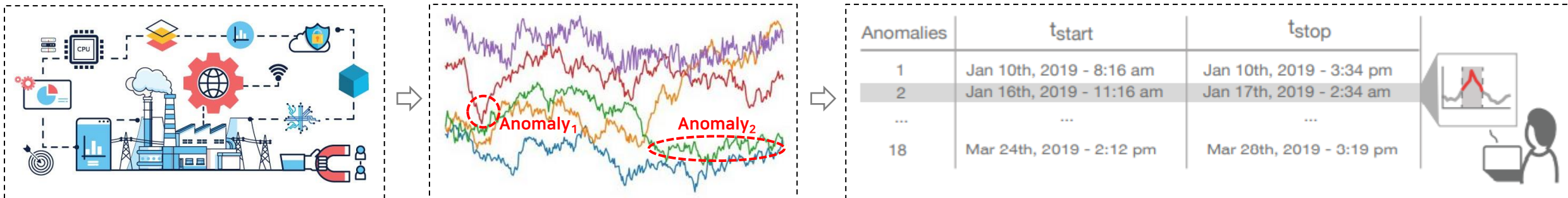
<Duration Anomaly>

Introduction

Anomaly Detection for Time Series

❖ 시계열 데이터에 대한 이상(Anomaly) 탐지의 어려움

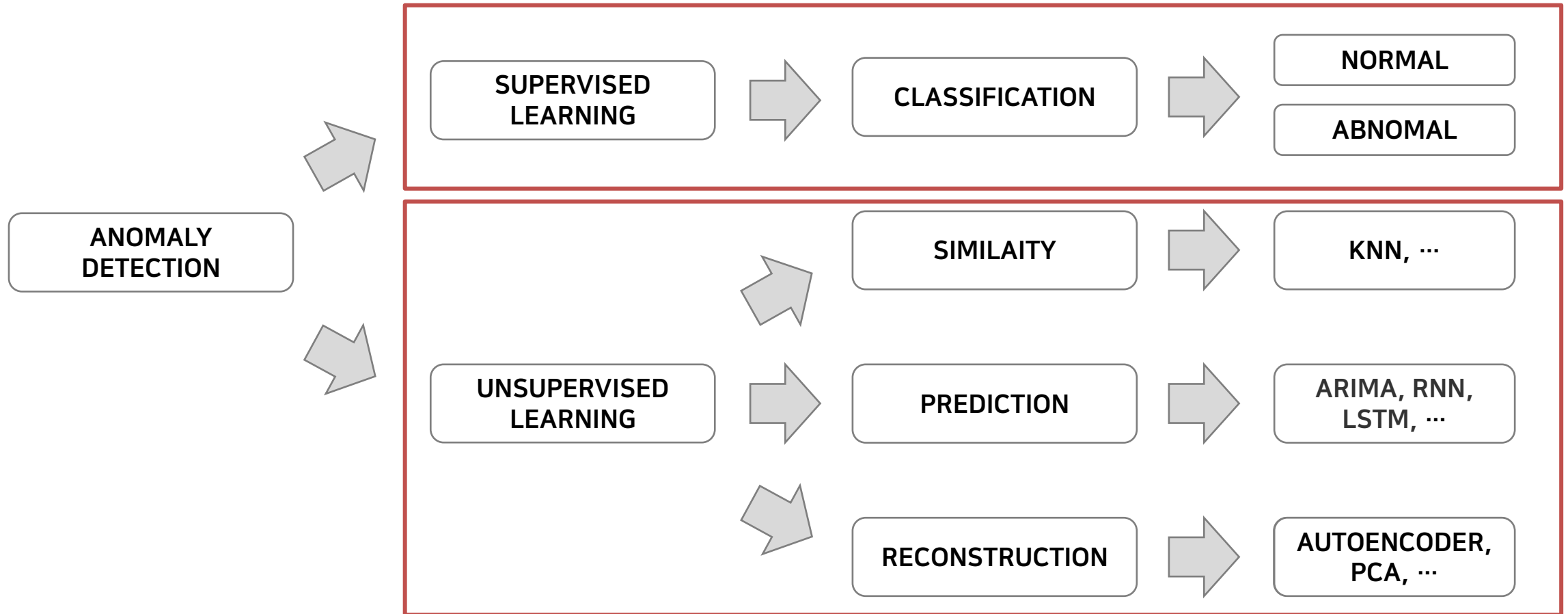
- 여러 비즈니스 영역에서 활용이 가능하기 때문에 발생할 수 있는 이상 유형이 다양
- 정상과 비정상 데이터를 정확히 구분하여 라벨링(Labeling)하기가 어려움
 - ✓ 이상 여부를 판별하기 위해서는 많은 양의 전문적인 도메인 지식을 요구됨
 - ✓ 시간과 비용이 많이 소요됨
 - ✓ 정상과 이상의 경계가 불분명함
- 정상 데이터에 비해 비정상 데이터의 비율이 훨씬 적기 때문에 데이터 불균형(Imbalanced data)이 발생 → 분류(Classification) 문제의 장애물



Introduction

Anomaly Detection for Time Series

❖ 시계열 데이터에 대한 이상 탐지 방법 도식화



목차

I. Introduction

- Anomaly Detection for Time Series

II. Autoencoder

III. Anomaly Detection for Time Series with Autoencoder

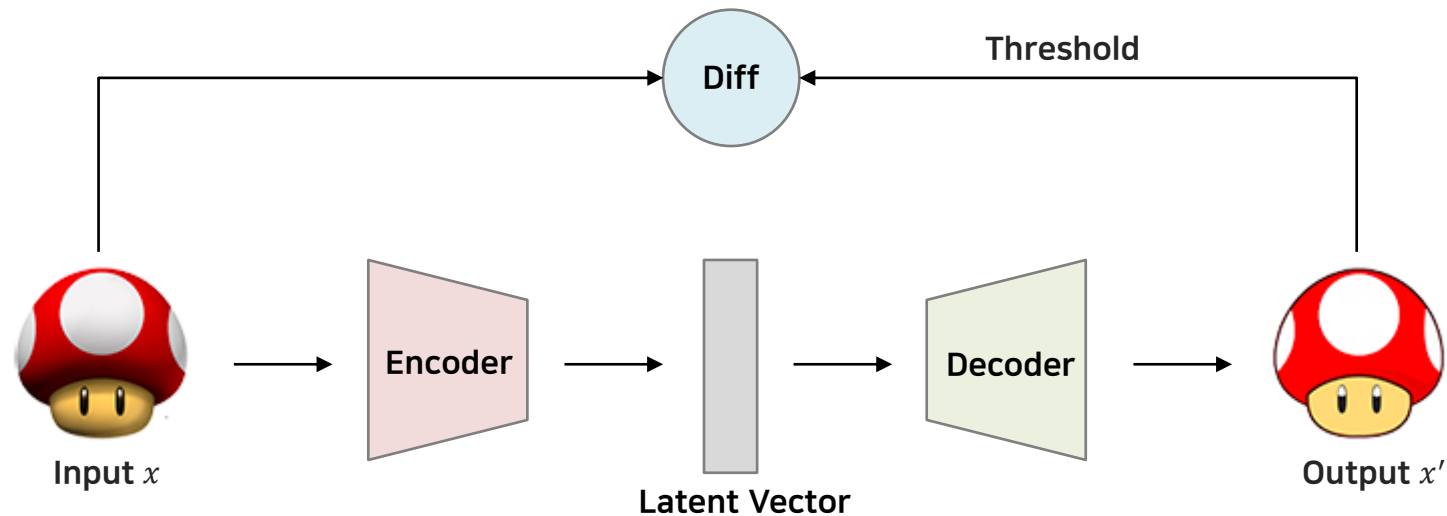
- LSTM-Autoencoder
- Autoencoder Ensembles
- Autoencoder within An Adversarial Training Framework

IV. Conclusion

Autoencoder

❖ Autoencoder 개념

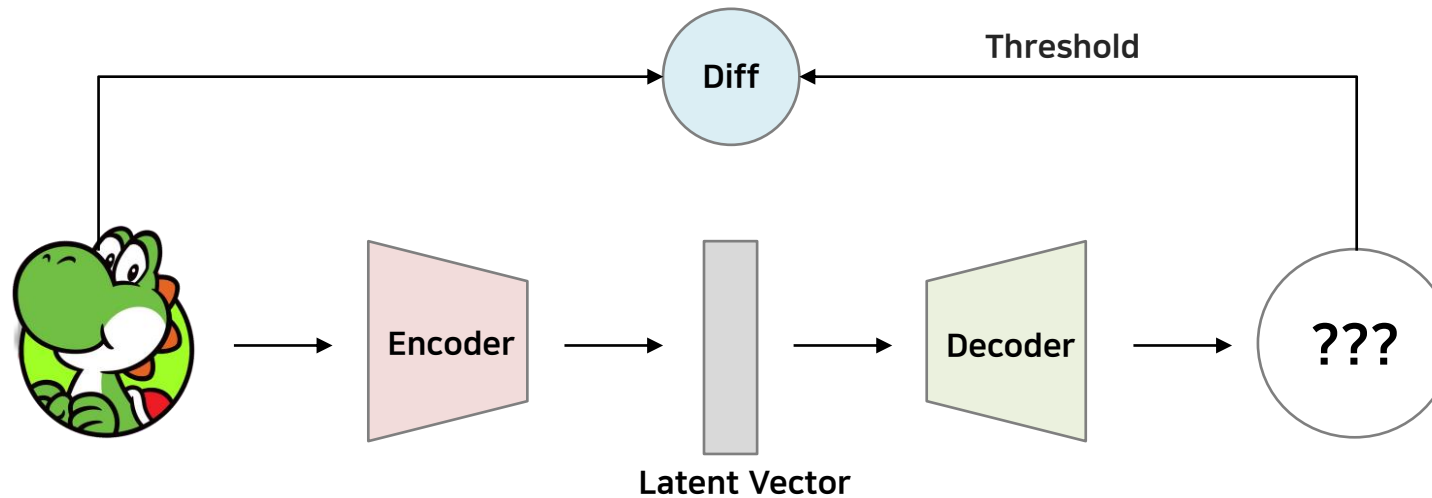
- 입력된 데이터를 재복원하는 비지도학습 방법론
- 입력된 값과 최대한 비슷하게 출력되도록 재구성하는 것이 목표
- 두 개의 구조로 구성: Encoder + Decoder
 - ✓ Encoder: 데이터를 압축하여 중요한 정보만 살리는 feature로 변환
 - ✓ Decoder: 압축된 feature를 이용하여 입력 데이터를 최대한 원본과 가깝게 복원



Autoencoder

❖ Autoencoder를 사용한 이상 탐지(Anomaly Detection)

- 입력 데이터로 정상 데이터만 사용
- Encoder의 입력 데이터와 Decoder의 결과인 복원 데이터의 차이를 줄이면서, Autoencoder는 정상 데이터의 특징 학습



- 비정상 데이터는 학습되지 않았기 때문에 비정상 데이터가 모델에 입력되면, 복원된 데이터가 입력과 많은 차이 발생
- 복원 에러(Reconstruction Error) = (입력 데이터) - (복원 데이터)
- 임계치(Threshold)를 정하고, 복원 에러가 임계치를 넘게 되면 이상 탐지(Anomaly detection)

목차

I. Introduction

- Anomaly Detection for Time Series

II. Autoencoder

III. Anomaly Detection for Time Series with Autoencoder

- LSTM-Autoencoder
- Autoencoder Ensembles
- Autoencoder within An Adversarial Training Framework

IV. Conclusion

LSTM-Autoencoder

Paper

❖ LSTM-based Encoder-Decoder for Multisensor Anomaly Detection

- Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., & Shroff, G. (2016). LSTM-based Encoder-Decoder for Multisensor Anomaly Detection. ICML, 2016. (608회 인용)

LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection

Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, Gautam Shroff
{MALHOTRA.PANKAJ, ANUSHA.RAMAKRISHNAN, GAURANGI.ANAND, LOVEKESH.VIG, PUNEET.A, GAUTAM.SHROFF}@TCS.COM
TCS Research, New Delhi, India

Abstract

Mechanical devices such as engines, vehicles, aircrafts, etc., are typically instrumented with numerous sensors to capture the behavior and health of the machine. However, there are often external factors or variables which are not captured by sensors leading to time-series which are inherently unpredictable. For instance, manual controls and/or unmonitored environmental conditions or load may lead to inherently unpredictable time-series. Detecting anomalies in

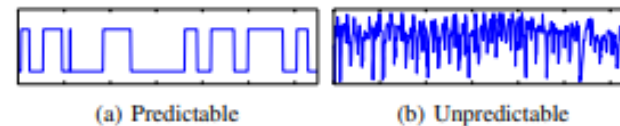


Figure 1. Readings for a manual control sensor.

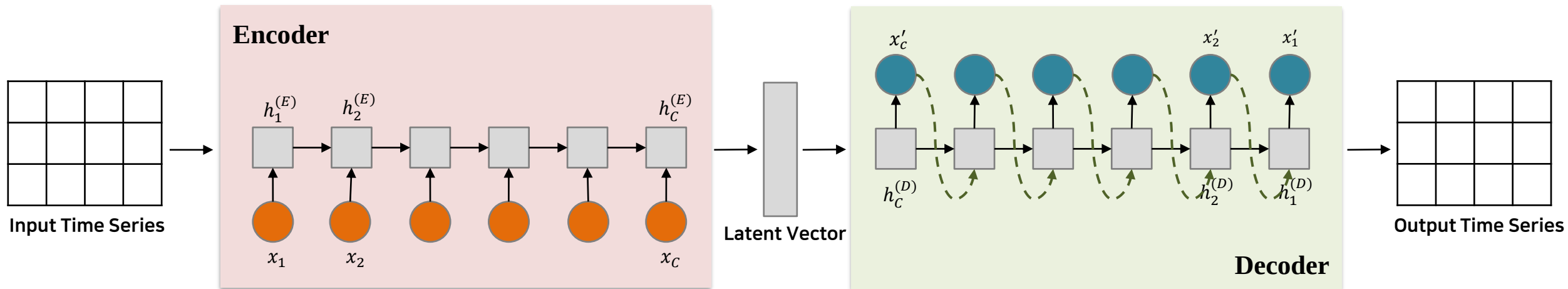
example, a laden machine behaves differently from an unladen machine. Further, the relevant information pertaining to whether a machine is laden or unladen may not be available. The amount of load on a machine at a time may be

LSTM-Autoencoder

Prerequisite Knowledge

❖ LSTM-Autoencoder

- Sequence 데이터를 위한 LSTM 구조를 사용하는 Autoencoder
- 시계열 데이터의 Anomaly Detection은 **시간적인 특성(Temporal Information)**을 고려
- 이전의 정보를 현재의 문제 해결에 활용할 수 있는 **LSTM 네트워크**를 이용



LSTM-Autoencoder

Method

❖ EncDec-AD¹

• Training(1)

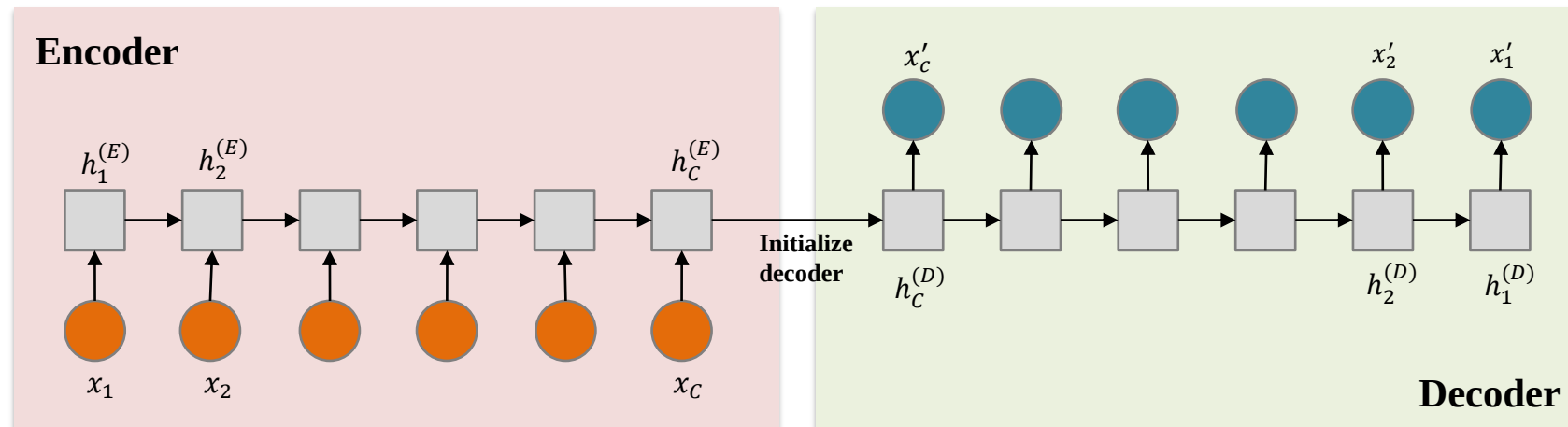
✓ 정상 데이터 사용

✓ Encoder

- 입력 데이터 = $\{x_1, x_2, \dots, x_C\}$, where $x_i \in R^m$, C : window size, m : # of variable
- Step $t = i$ 에서 Hidden vector는 이전 단계의 Hidden vector($h_{i-1}^{(E)}$)와 입력값(x_i)을 활용하여 구함.

$$h_i^{(E)} = f(x_i, h_{i-1}^{(E)})$$

- 마지막 Hidden vector가 Latent vector가 되어 Decoder 단계의 최초 Hidden vector로 사용



- x_i : the input vector at time t_i
- x'_i : the predicted vector at time t_i
- $h_i^{(E)}$: hidden vector of encoder at time t_i
- $h_i^{(D)}$: hidden vector of decoder at time t_i
- C : window size. the number of LSTM unit.
- S_N : set of normal training sequences

LSTM-Autoencoder

Method

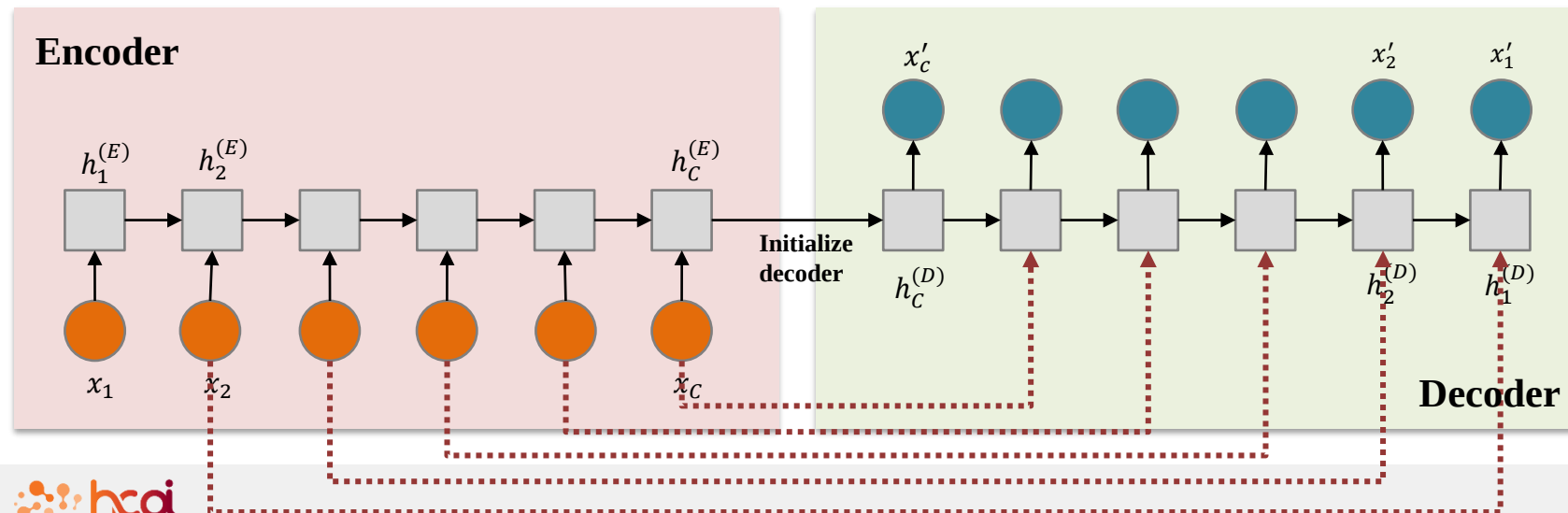
❖ EncDec-AD

- Training(2)

- ✓ Decoder

- 입력 데이터 = $\{x_C, \dots, x_2, x_1\}$, where $x_i \in R^m$, C : window size, m : # of variable
 - Encoder의 입력 데이터를 역순으로 Reconstruction
 - Step $t = (i - 1)$ 의 Hidden vector는 이전 단계의 Hidden vector($h_i^{(D)}$)와 입력값(x_i)을 활용하여 구함.

$$h_{i-1}^{(D)} = g(x_i, h_i^{(D)})$$



- x_i : the input vector at time t_i
- x'_i : the predicted vector at time t_i
- $h_i^{(E)}$: hidden vector of encoder at time t_i
- $h_i^{(D)}$: hidden vector of decoder at time t_i
- C : window size. the number of LSTM unit.
- S_N : set of normal training sequences

LSTM-Autoencoder

Method

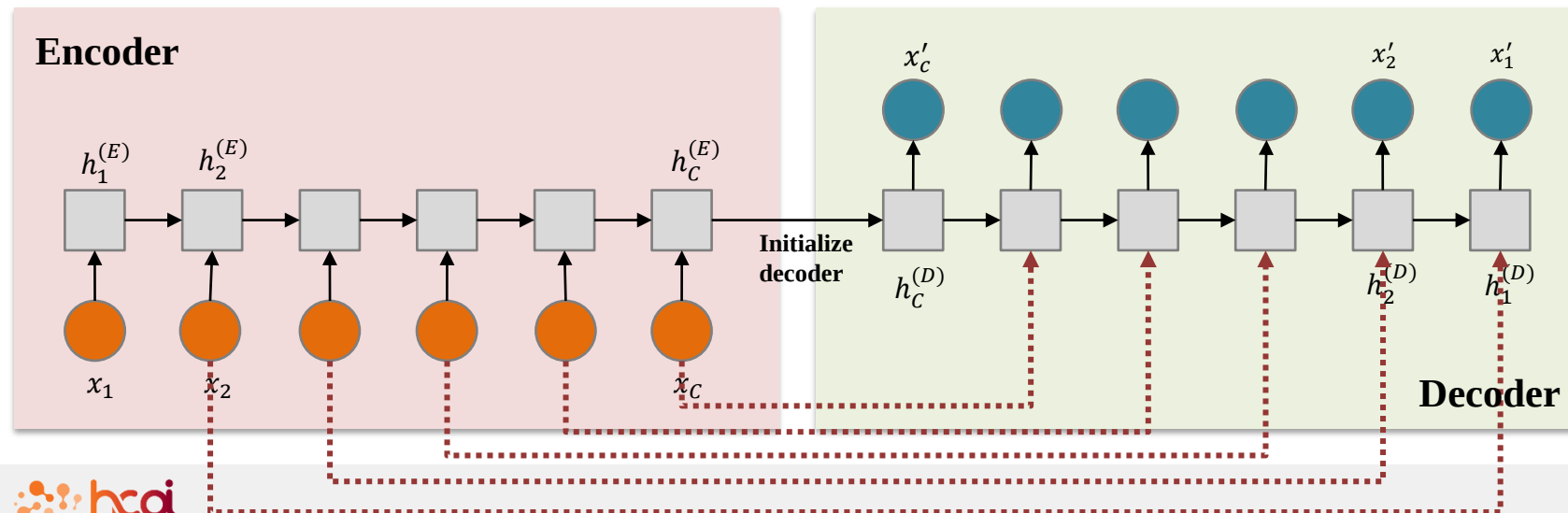
❖ EncDec-AD

- Training(3)

- ✓ Objective Function

- Encoder의 입력값(x_i)과 Decoder의 출력값(x'_i)의 차이에 대한 MSE(Mean Squared Error)를 최소화 하도록 학습

$$\text{minimize } \sum_{X \in S_N} \sum_{i=1}^L \|x_i - x'_i\|^2$$



- x_i : the input vector at time t_i
- x'_i : the predicted vector at time t_i
- $h_i^{(E)}$: hidden vector of encoder at time t_i
- $h_i^{(D)}$: hidden vector of decoder at time t_i
- C : window size. the number of LSTM unit.
- S_N : set of normal training sequences

LSTM-Autoencoder

Method

❖ EncDec-AD

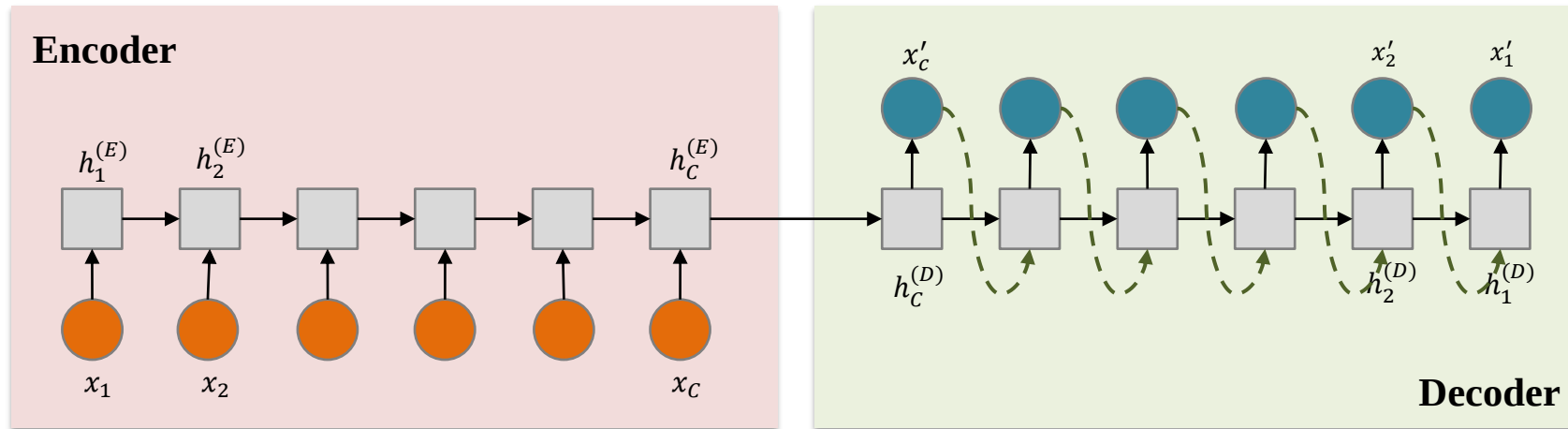
- Inference

- ✓ Training 과정과 대체로 비슷

- ✓ Decoder 단계

- ✓ Step $t = (i - 1)$ 의 Hidden vector는 이전 단계의 Hidden vector($h_i^{(D)}$)와 **Decoder의 출력값(x'_i)을 활용하여 구함.**

$$h_{i-1}^{(D)} = g(x'_i, h_i^{(D)})$$



- x_i : the input vector at time t_i
- x'_i : the predicted vector at time t_i
- $h_i^{(E)}$: hidden vector of encoder at time t_i
- $h_i^{(D)}$: hidden vector of decoder at time t_i
- C : window size. the number of LSTM unit.
- S_N : set of normal training sequences

LSTM-Autoencoder

Method

❖ Computing Likelihood of Anomaly



- Reconstruction Error

- ✓ v_{N1} 데이터를 이용하여 error vector($e^{(i)}$)를 구함

$$e^{(i)} = |x^{(i)} - x'_i|$$

- ✓ Error vector를 사용하여 정규분포 $N(\mu, \Sigma)$ 의 μ, Σ 추론

- Anomaly Score

- ✓ $e^{(i)}$ 에서 정규분포 $N(\mu, \Sigma)$ 까지의 거리(마할라노비스 거리). 거리가 멀어질 수록 이상값으로 판단

$$a^{(i)} = (e^{(i)} - \mu)^T \Sigma^{-1} (e^{(i)} - \mu)$$

- ✓ 임계치(τ)를 정하고, 그 값보다 Anomaly Score가 크면 이상치로 판단

$$\begin{cases} \text{if } a^{(i)} > \tau, & \text{anomalous} \\ \text{if } a^{(i)} \leq \tau, & \text{normal} \end{cases}$$

LSTM-Autoencoder

Experiments and Results

❖ Datasets

Dataset	Predictable	Dimension	Periodicity
Power demand	Yes	1	Periodic
Space shuttle	Yes	1	Periodic
Engine-P	Yes	12	Aperiodic
Engine-NP	No	12	Aperiodic
ECG	Yes	1	Quasi-periodic

❖ Evaluation Metrics

- F_β score 사용

$$F_\beta = \frac{(1 + \beta^2) \times Precision \times Recall}{\beta^2 \times Precision + Recall}$$

- ✓ $\beta = 1$ 이면, F_β score = F_1 score
- ✓ $\beta > 1$ 이면, Recall에 가중치
- ✓ $\beta < 1$ 이면, Precision 에 가중치
- $\beta < 1$ 로 가정
 - ✓ Positive: 비정상(anomalous) / Negative: 정상(normal)
 - ✓ Sequence에서 비정상 비율이 높지 않아 Recall이 낮을 수 있기 때문에 Recall 에 가중치를 취함

LSTM-Autoencoder

Experiments and Results

❖ Result

- 모든 데이터 셋에서 TPR/FPR 값이 1.0 보다 높음 → 비정상 데이터를 잘 탐지 할 수 있음
- 다양한 **주기적(Periodic)** 시계열 데이터에 대해서도 좋은 성능을 나타냄
- Sequence를 **예측할 수 없는 데이터 셋에 대해 성능이 좋음**(ex.Engine-NP). 다만 예측 가능한 데이터 셋에 대해 시계열 예측(prediction) 기반 이상 감지 모델 LSTM-AD보다 좋은 성능을 내지 못함

Dataset	Predictable	Periodicity	EncDec-AD			LSTM-AD ¹	
			β	F_{β} score	TPR/FPR	β	F_{β} score
Power demand	Yes	Periodic	0.1	0.77	33.0	0.1	0.90
Space shuttle	Yes	Periodic	0.05	0.81	4.9	0.1	0.84
Engine-P	Yes	Aperiodic	0.05	0.82	13.8	0.1	0.89
Engine-NP	No	Aperiodic	0.05	0.83	∞	0.05	0.03
ECG	Yes	Quasi-periodic	0.05	0.65	∞	X	X

¹Malhotra, Pankaj, Vig, Lovekesh, Shroff, Gautam, and Agarwal, Puneet. Long short term memory networks for anomaly detection in time series. In ESANN, 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2015

목차

I. Introduction

- Anomaly Detection for Time Series

II. Autoencoder

III. Anomaly Detection for Time Series with Autoencoder

- LSTM-Autoencoder
- Autoencoder Ensembles
- Autoencoder within An Adversarial Training Framework

IV. Conclusion

Autoencoder Ensembles

Paper

❖ Outlier Detection for Time Series with Recurrent Autoencoder Ensembles

- KIEU, Tung, et al. Outlier Detection for Time Series with Recurrent Autoencoder Ensembles. In: IJCAI. 2019 (110회 인용)

Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)

Outlier Detection for Time Series with Recurrent Autoencoder Ensembles

Tung Kieu, Bin Yang*, Chenjuan Guo and Christian S. Jensen

Department of Computer Science, Aalborg University, Denmark

{tungkvt, byang, cguo, csj}@cs.aau.dk

Abstract

We propose two solutions to outlier detection in time series based on recurrent autoencoder ensembles. The solutions exploit autoencoders built using sparsely-connected recurrent neural networks (S-RNNs). Such networks make it possible to generate multiple autoencoders with different neural network connection structures. The two solutions are ensemble frameworks, specifically an independent framework and a shared framework, both of which combine multiple S-RNN based autoencoders to enable outlier detection. This ensemble-based approach aims to reduce the effects of some au-

observations that differ significantly from the remaining observations [Aggarwal, 2013]. Such outliers can offer important insights. For example, outliers in ECG time series may indicate potential heart attacks while the remaining observations represent normal physical conditions [Chia and Syed, 2014]. Further, we consider outlier detection in an *unsupervised* setting, where we do not rely on prior labels that indicate outliers. Rather, we aim at identifying outliers in unlabeled time series.

Most existing methods for outlier detection are based on similarity search [Yeh *et al.*, 2016] and density-based clustering [Breunig *et al.*, 2000]. Recently, neural network based autoencoders [Xia *et al.*, 2015; Chen *et al.*, 2017; Kieu *et al.*, 2018b] are proposed for the detection of out-

Autoencoder Ensembles

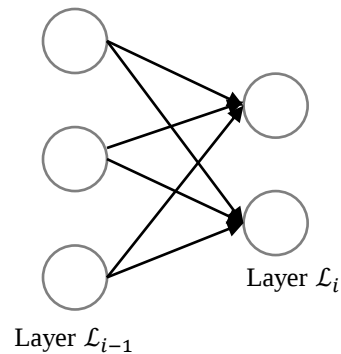
Contributions

❖ Autoencoder 모델의 한계

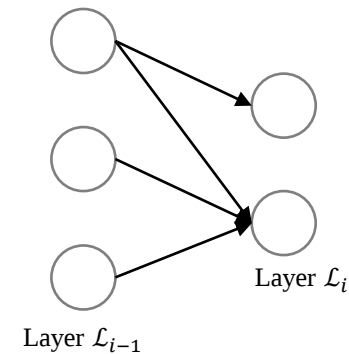
- 정상데이터만으로 학습하기 때문에, 다른 데이터가 들어와도 Training set과 비슷하게 만드는 과적합(Overfitting) 문제 발생
 - ✓ 과적합(Overfitting): 학습 모델이 주어진 데이터에 너무 과하게 맞춰져서, 조금이라도 다른 데이터만 들어와도 다른 결과로 예측하여 결과적으로 정확도가 낮아지는 현상

❖ 제안 모델

- Single Autoencoder가 아닌 서로 다른 네트워크 구조를 가진 여러개의 Autoencoder 결합
- 서로 다른 네트워크 구조를 Autoencoder를 생성하기 위해 Sparsely-connected Autoencoder 적용



(a) Fully-connected layers



(b) Sparsely-connected layers

Autoencoder Ensembles

Method

❖ Sparsely-connected RNNs(S-RNNs)

- 서로 다른 구조를 가진 Autoencoder를 여러개 생성하기 위해 RNN 구조 변경

1. RNN unit 사이의 연결을 임의 제거

- ✓ Training에 문제가 발생

2. RNN unit의 사이의 연결을 임의 추가

- ✓ Recurrent Skip Connection Networks (RSCNs)¹ 을 참고

- Sparsely-connected RNNs(S-RNNs)

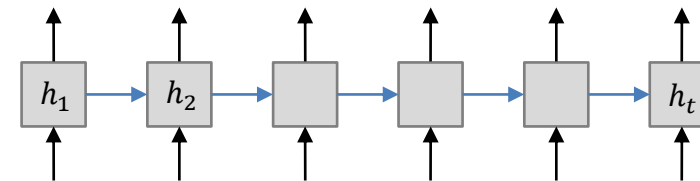
$$h_t = \frac{f(s_t, h_{t-1}) \cdot w_t^{(f)} + f'(s_t, h_{t-L}) \cdot w_t^{(f')}}{\|w_t\|_0}$$

- ✓ Time step $t - 1$ 과 $t - L$ 의 hidden state를 임의로 연결한 hidden state

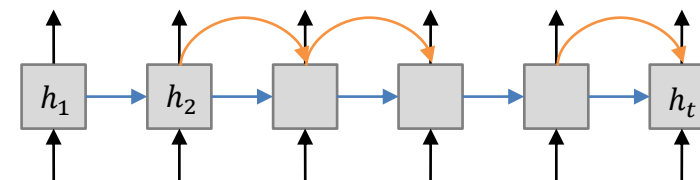
- $f(\cdot)$: step $t - 1$ 의 hidden state로 측정된 non-linear function
- $f'(\cdot)$: step $t - L$ 의 hidden state로 측정된 non-linear function
- w_t : step t 의 connection을 결정하는 sparseness weight vector

$$w_t^{(f)} \in \{0,1\} \text{ and } w_t^{(f')} \in \{0,1\}, \text{ 단 모두 0은 아님}$$

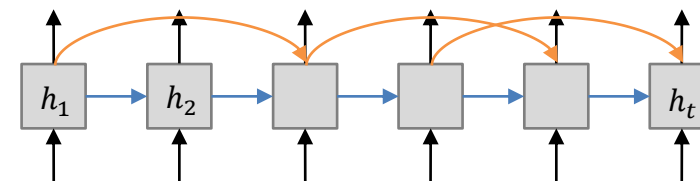
- $\|w_t\|_0$: sparseness weight vector가 0이 아닌 element 갯수. 단, $\|w_t\|_0 \neq 0$



(a) RNN



(b) S-RNN, L=1



(b) S-RNN, L=2

Autoencoder Ensembles

Method

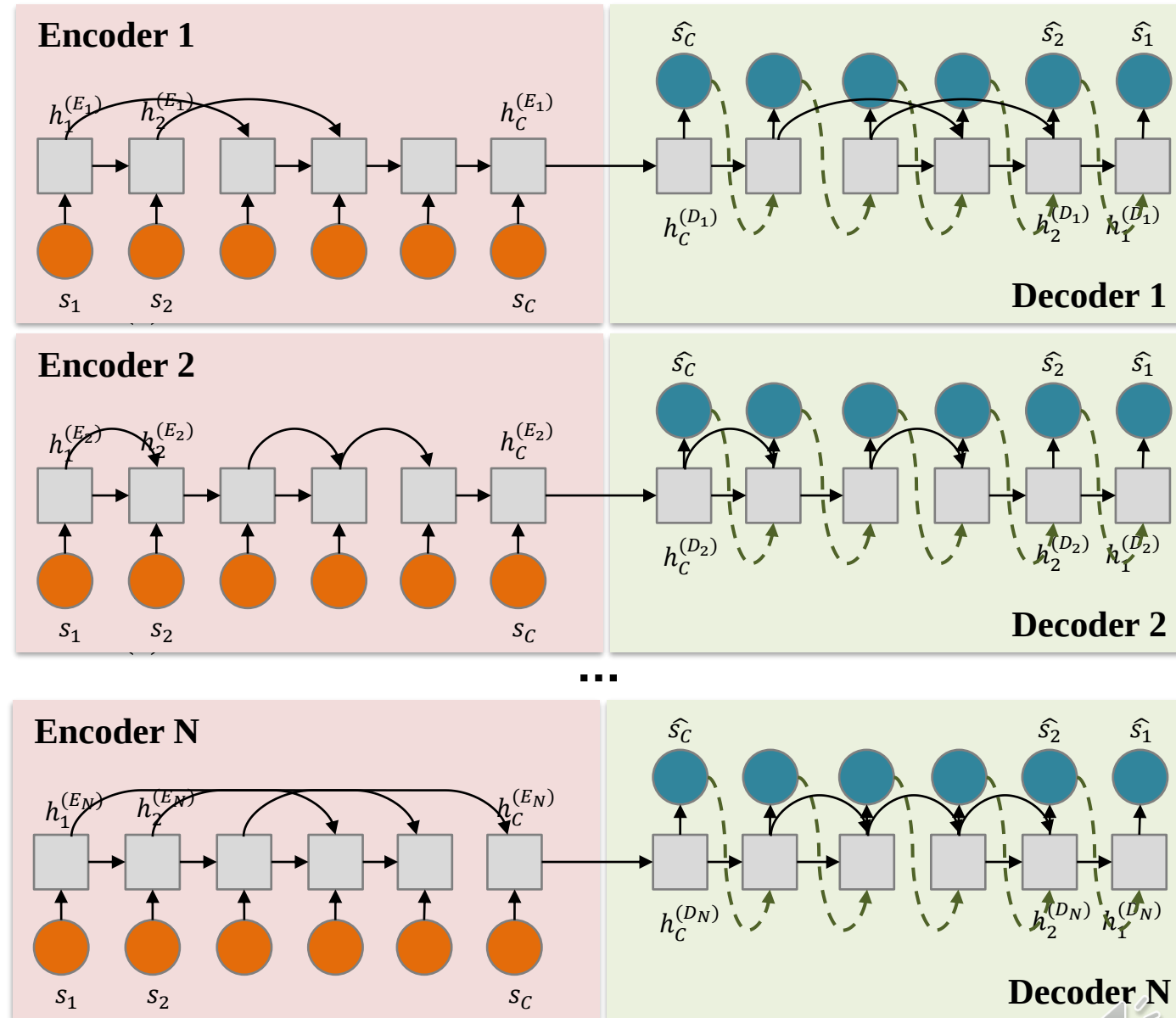
❖ S-RNN Autoencoder Ensembles(1)

- Independent Framework(IF)
- ✓ 구조가 다른 N 개의 S-RNN Autoencoder를 독립적으로 훈련

✓ Objective Function

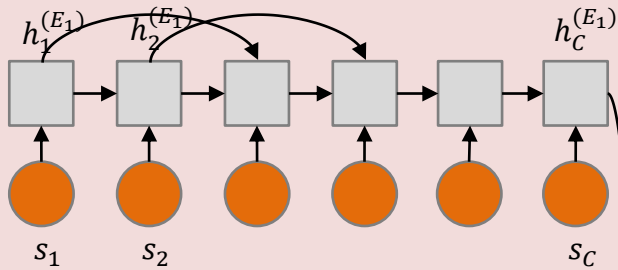
$$\text{minimize } J_i = \sum_{t=1}^C \|s_t - \hat{s}_t^{(D_i)}\|_2^2$$

- $\hat{s}_t^{(D_i)}$: reconstructed vector at time step t from decoder D_i
- $\|\cdot\|_2$: L2-norm of a vector

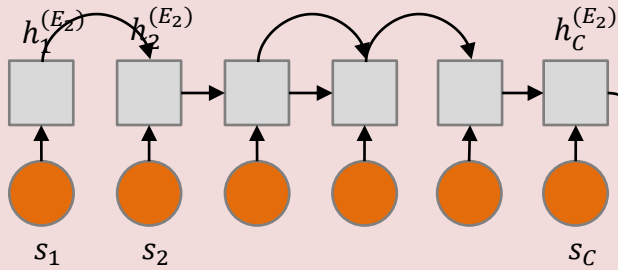


Autoencoder Ensembles

Encoder 1

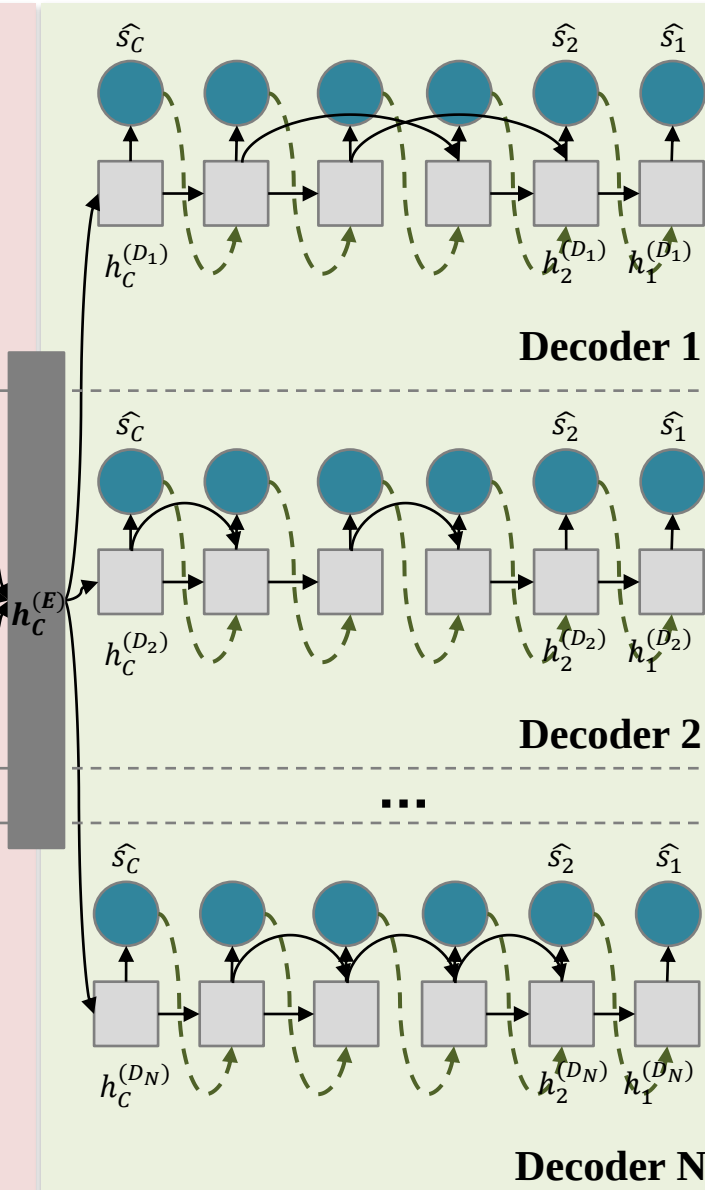
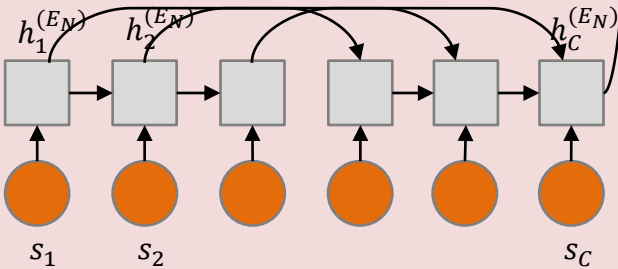


Encoder 2



...

Encoder N



❖ S-RNN Autoencoder Ensembles(2)

• Shared Framework(SF)

- ✓ 구조가 다른 N 개의 S-RNN Autoencoder가 상호작용하며 훈련

✓ Shared layer

$$h_c^{(E)} = \text{concatenate}(h_c^{(E_1)} \cdot W^{E_1}, \dots, h_c^{(E_N)} \cdot W^{E_N})$$

- Shared layer를 통해서 N 개의 S-RNN Autoencoder가 상호작용
- 모든 Encoder의 마지막 Hidden state의 Linear 결합
- W^{E_i} : linear weight matrices

✓ Objective Function

$$\begin{aligned} \text{minimize } J &= \sum_{i=1}^N J_i + \lambda \|h_c^{(E)}\|_1 \\ &= \sum_{i=1}^N \sum_{t=1}^C \|s_t - \hat{s}_t^{(D_i)}\|_2^2 + \lambda \|h_c^{(E)}\|_1 \end{aligned}$$

- λ : L1 regularization($\|h_c^{(E)}\|_1$)의 중요도를 조절하는 가중치

Autoencoder Ensembles

Method

❖ S-RNN Autoencoder Ensembles

- Autoencoder Ensemble Anomaly Scoring

- ✓ Autoencoder Ensembles Framework에서 Time step t vector의 Anomaly score를 계산

- ✓ Reconstruction error

- Time step $t = k$ 일 때, 모든 Autoencoder N 의 Reconstruction error

$$\left\{ \|s_k - \hat{s}_k^{(1)}\|_2^2, \|s_k - \hat{s}_k^{(2)}\|_2^2, \dots, \|s_k - \hat{s}_k^{(N)}\|_2^2 \right\}$$

- ✓ Anomaly score

$$Anomaly\ score(s_k) = median\left(\|s_k - \hat{s}_k^{(1)}\|_2^2, \|s_k - \hat{s}_k^{(2)}\|_2^2, \dots, \|s_k - \hat{s}_k^{(N)}\|_2^2\right)$$

- *median*을 사용함으로써 Overfitting 된 Autoencoder의 영향 감소

Autoencoder Ensembles

Experiments and Results

❖ Datasets

Dataset	Description	Dimension	# of Sub dataset	# of Observations
NAB	Numenta Anomaly Benchmark	1	10	5,000~2,0000
ECG	Electrocardiography	3	7	3,750~5,400

❖ Evaluation Metrics

Evaluation Metrics	Description
PR-AUC	• 불균형 데이터 • Positive class를 탐지하는 중요도가 Negative class 를 탐지하는 중요도보다 높을때 • 예) 암 환자 진단
ROC-AUC	• 균형 데이터 • Positive class 탐지와 Negative class 탐지의 중요도가 비슷할 때 • 예) 개와 고양이 분류

Autoencoder Ensembles

Experiments and Results

❖ Existing Solutions

Method	Description
LOF	Local Outlier Factor Density-based anomaly detection 모델
SVM	Support Vector Machines Kernel-based method
ISF	Isolation Forest
MP	Matrix Profile I. Similarity-based anomaly detection 의 SOTA 모델
RN	Rand-Net. Non-sequential data 대한 Autoencoder ensembles를 적용한 SOTA 모델
CNN	CNN-based Autoencoder 모델
LSTM	LSTM-based Autoencoder 모델 Time series data에 대한 anomaly detection 의 SOTA 모델
RSCN	LSTM units을 RSCN units으로 교체한 모델

Autoencoder Ensembles

Experiments and Results

❖ Overall Accuracy

- 평가 지표 관점
 - ✓ PR-AUC : deep learning 모델과 아닌 모델의 차이가 거의 없음
 - ✓ ROC-AUC: deep learning 모델이 대체로 좋은 성능을 냄
- Sequence 데이터 관점: Rand-Net(RN)보다 제안모델이 더 좋은 성능 → 제안 모델이 Sequence 데이터에 더 적합
- 관측 변수 관점: 제안 모델이 다른 모델에 대해서 대체로 성능이 좋고, **단변량**과 **다변량** 모두에서 성능이 뛰어남
- 제안 방법론 관점: Shared Framework(SF)가 Independent Framework(IF)보다 좀 더 좋은 성능을 냄

	Data Set	PR-AUC										ROC-AUC									
		SVM	LOF	ISF	MP	RN	CNN	LSTM	RSCN	IF	SF	SVM	LOF	ISF	MP	RN	CNN	LSTM	RSCN	IF	SF
NAB, Univariate	Cloudwatch	0.908	0.904	0.909	0.892	0.886	0.889	0.886	0.895	0.885	0.912	0.566	0.516	0.559	0.563	0.569	0.632	0.662	0.634	0.675	0.667
	Anomaly	0.902	0.902	0.911	0.894	0.902	0.926	0.904	0.915	0.912	0.917	0.551	0.522	0.627	0.613	0.629	0.657	0.753	0.755	0.761	0.777
	Traffic	0.908	0.905	0.914	0.913	0.895	0.919	0.859	0.865	0.922	0.914	0.551	0.584	0.566	0.548	0.568	0.608	0.625	0.667	0.645	0.689
	Tweets	0.904	0.902	0.911	0.901	0.898	0.907	0.876	0.893	0.915	0.907	0.538	0.527	0.557	0.533	0.564	0.571	0.554	0.599	0.608	0.653
	AdExchange	0.898	0.897	0.902	0.897	0.891	0.881	0.882	0.893	0.893	0.903	0.551	0.513	0.539	0.567	0.565	0.602	0.574	0.603	0.691	0.646
	KnownCause	0.929	0.905	0.915	0.893	0.889	0.891	0.872	0.916	0.916	0.987	0.639	0.517	0.657	0.584	0.612	0.613	0.642	0.609	0.638	0.621
	Average	0.908	0.903	0.910	0.898	0.894	0.902	0.880	0.896	0.907	0.923	0.566	0.530	0.584	0.568	0.585	0.614	0.635	0.645	0.670	0.676
ECG, Multivariate	chf01	0.912	0.923	0.886	0.866	0.943	0.967	0.904	0.971	0.961	0.958	0.629	0.521	0.508	0.815	0.628	0.714	0.646	0.702	0.681	0.689
	chf13	0.977	0.964	0.975	0.936	0.941	0.944	0.938	0.966	0.963	0.967	0.618	0.577	0.658	0.644	0.661	0.642	0.581	0.562	0.823	0.803
	ltstdb240	0.984	0.963	0.991	0.906	0.932	0.980	0.947	0.973	0.987	0.995	0.678	0.530	0.798	0.875	0.727	0.747	0.651	0.587	0.753	0.879
	ltstdb43	0.982	0.959	0.980	0.964	0.962	0.974	0.956	0.966	0.982	0.985	0.617	0.531	0.612	0.512	0.558	0.561	0.589	0.581	0.709	0.717
	mitdb180	0.953	0.969	0.944	0.958	0.955	0.985	0.974	0.973	0.988	0.980	0.720	0.539	0.694	0.654	0.673	0.670	0.610	0.580	0.678	0.694
	stdb308	0.879	0.955	0.897	0.902	0.911	0.978	0.977	0.981	0.983	0.987	0.821	0.546	0.722	0.820	0.786	0.639	0.735	0.842	0.856	0.838
	xmitdb108	0.949	0.934	0.901	0.888	0.949	0.963	0.862	0.845	0.964	0.957	0.516	0.539	0.653	0.728	0.603	0.796	0.816	0.813	0.785	0.822
	Average	0.948	0.952	0.939	0.917	0.942	0.970	0.937	0.954	0.975	0.976	0.657	0.540	0.664	0.721	0.662	0.681	0.661	0.667	0.755	0.777

Autoencoder Ensembles

Experiments and Results

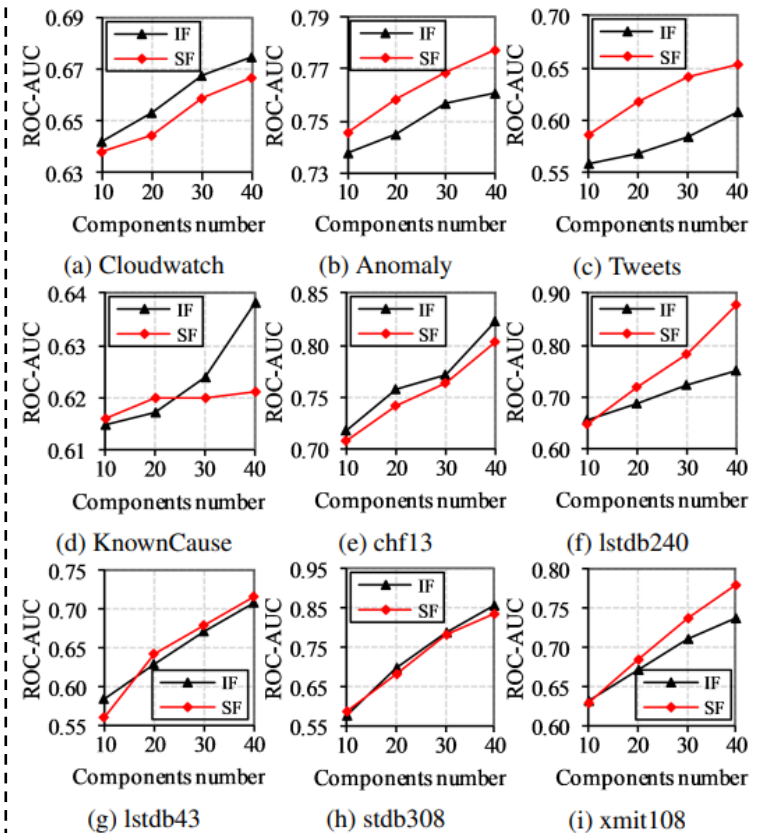
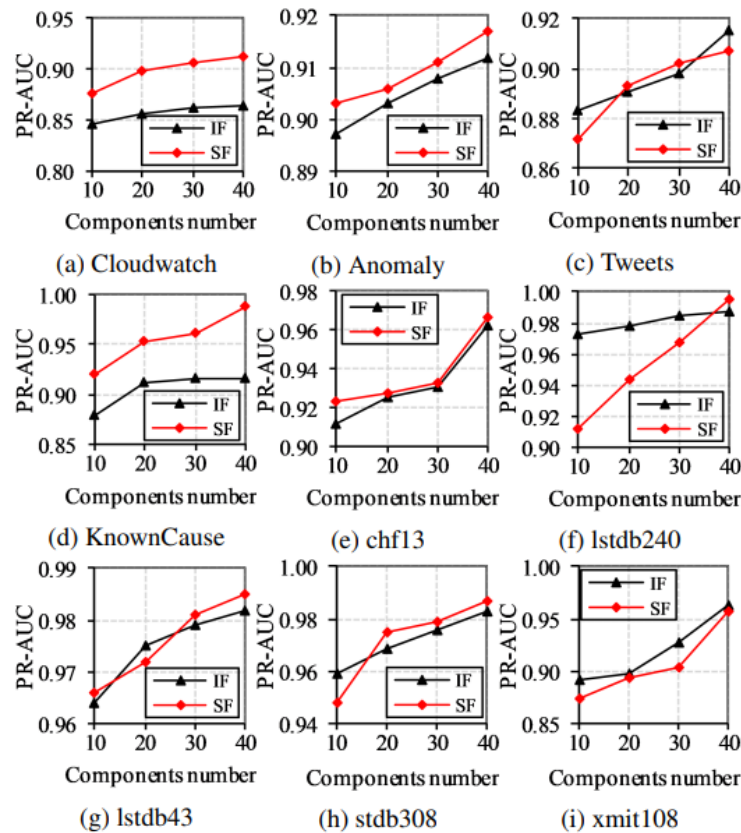
❖ Effect of N

- Setting

- ✓ $N = \{10, 20, 30, 40\}$
- ✓ 4 *NAB* datasets & 5 *ECG* datasets

- Result

- ✓ Independent Framework(IF)와 Shared Framework(SF) 에서 N 이 커질수록 성능이 향상
- ✓ 다양한 구조를 가진 Autoencoder가 많아질 수록 정확도 향상에 도움이 됨

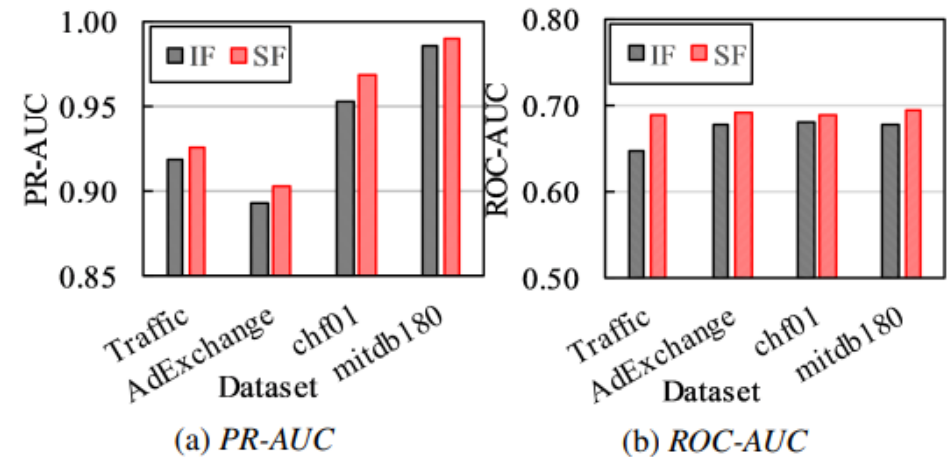


Autoencoder Ensembles

Outlier Detection for Time Series with Recurrent Autoencoder Ensembles

❖ IF vs. SF

- S-RNNs을 임의로 생성하기 때문에 Independent Framework(IF)와 Shared Framework(SF)의 구조가 다름
- 두 Framework의 정확한 성능 확인을 위해 S-RNNs 구조를 동일하게 실험
- Setting
 - ✓ $N = 40$
 - ✓ 2 *NAB* datasets & 2 *ECG* datasets
- Result
 - ✓ Shared Framework이 학습에 효과가 큼
 - ✓ Shared Framework은 Encoder의 마지막 Hidden states를 결합하여 Shared layer를 생성해야 하므로 많은 memory 사용함

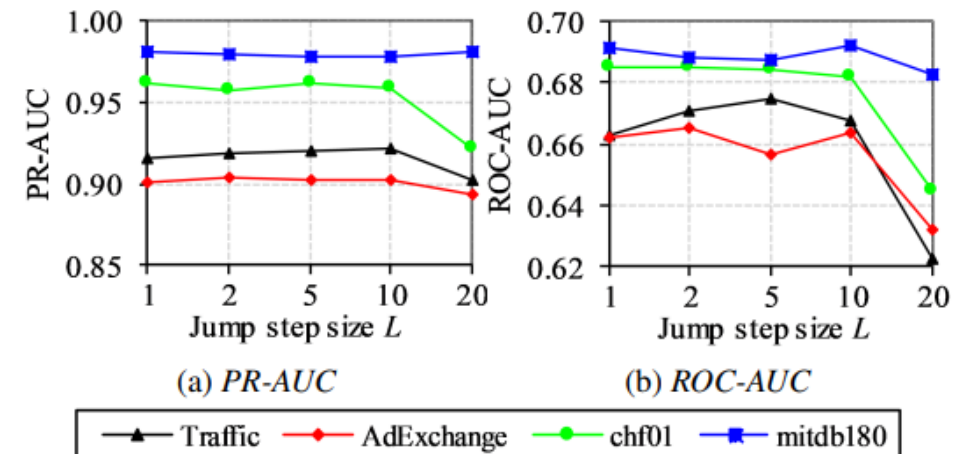


Autoencoder Ensembles

Outlier Detection for Time Series with Recurrent Autoencoder Ensembles

❖ Effect of L

- L : Hidden state를 연결하는 jump step size
- Setting
 - ✓ $L = \{1, 2, 5, 10, 20\}$
 - ✓ 2 *NAB* datasets & 2 *ECG* datasets
- Result
 - ✓ $L = \{1, 2, 5, 10\}$ 일 때 정확도에 영향이 없음
 - 작은 Jump step 에 민감하지 않음
 - ✓ $L = 20$ 일 때, 정확도 감소
 - Jump step이 커지면, 중요한 시간 정보가 손실됨
 - 정확도에 부정적인 영향을 미치는 쓸모 없는 과거 정보를 사용



목차

I. Introduction

- Anomaly Detection for Time Series

II. Autoencoder

III. Anomaly Detection for Time Series with Autoencoder

- LSTM-Autoencoder
- Autoencoder Ensembles
- Autoencoder within An Adversarial Training Framework

IV. Conclusion

Autoencoder within An Adversarial Training Framework

Paper

❖ USAD : UnSupervised Anomaly Detection on Multivariate Time Series

- AUDIBERT, Julien, et al. USAD: unsupervised anomaly detection on multivariate time series. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020. (60회 인용)

Applied Data Science Track Paper

KDD '20, August 23–27, 2020, Virtual Event, USA

USAD : UnSupervised Anomaly Detection on Multivariate Time Series

Julien Audibert
julien.audibert@orange.com
EURECOM
Biot, France
Orange
Sophia Antipolis, France

Pietro Michiardi
pietro.michiardi@eurecom.fr
EURECOM
Biot, France

Frédéric Guyard
frederic.guyard@orange.com
Orange Labs
Sophia Antipolis, France

Sébastien Marti
sebastien.marti@orange.com
Orange
Sophia Antipolis, France

Maria A. Zuluaga
zuluaga@eurecom.fr
EURECOM
Biot, France

ABSTRACT

The automatic supervision of IT systems is a current challenge at Orange. Given the size and complexity reached by its IT operations, the number of sensors needed to obtain measurements over time, used to infer normal and abnormal behaviors, has increased dramatically making traditional expert-based supervision methods

Event, CA, USA. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3394486.3403392>

1 INTRODUCTION

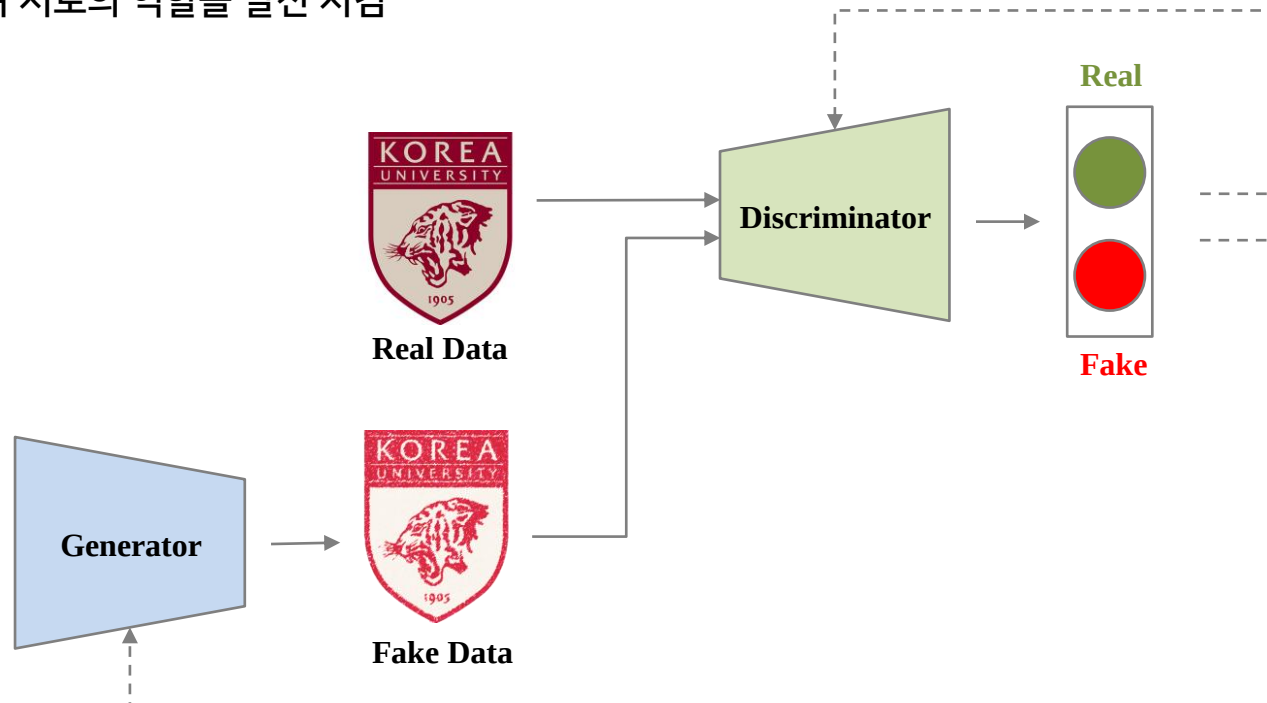
IT system monitoring is a supervision process on measurable events and outputs of a system, which is used as a reference specifying

Autoencoder within An Adversarial Training Framework

Prerequisite Knowledge

❖ Generative Adversarial Networks(GANs)

- GANs 기본 구조: **생성기(Generator) + 판별기(Discriminator)**
 - ✓ 생성기: 원본 데이터(Real data)와 유사한 가짜 데이터(Fake data) 생성
 - ✓ 판별기: 원본 데이터와 가짜 데이터를 구별
- 두 네트워크가 경쟁하며 서로의 역할을 발전 시킴



Autoencoder within An Adversarial Training Framework

Contributions

❖ 기존 이상탐지 모델의 한계

- Autoencoder
 - ✓ 비정상 데이터가 정상 데이터와 비슷하면, reconstruction error가 작아서 이상 감지가 어려움
- GANs
 - ✓ 생성기(Generator)와 분류기(Discriminator)의 불균형. 안정적인 학습이 어려움
 - ✓ Mode-Collapse, Non-Convergence 문제 발생
 - Mode-Collapse: 생성기(Generator)가 제한된 종류의 샘플 생성
 - Non-Convergence: Model의 Parameter가 수렴하지 않고, 불안정

❖ 제안 모델의 이점

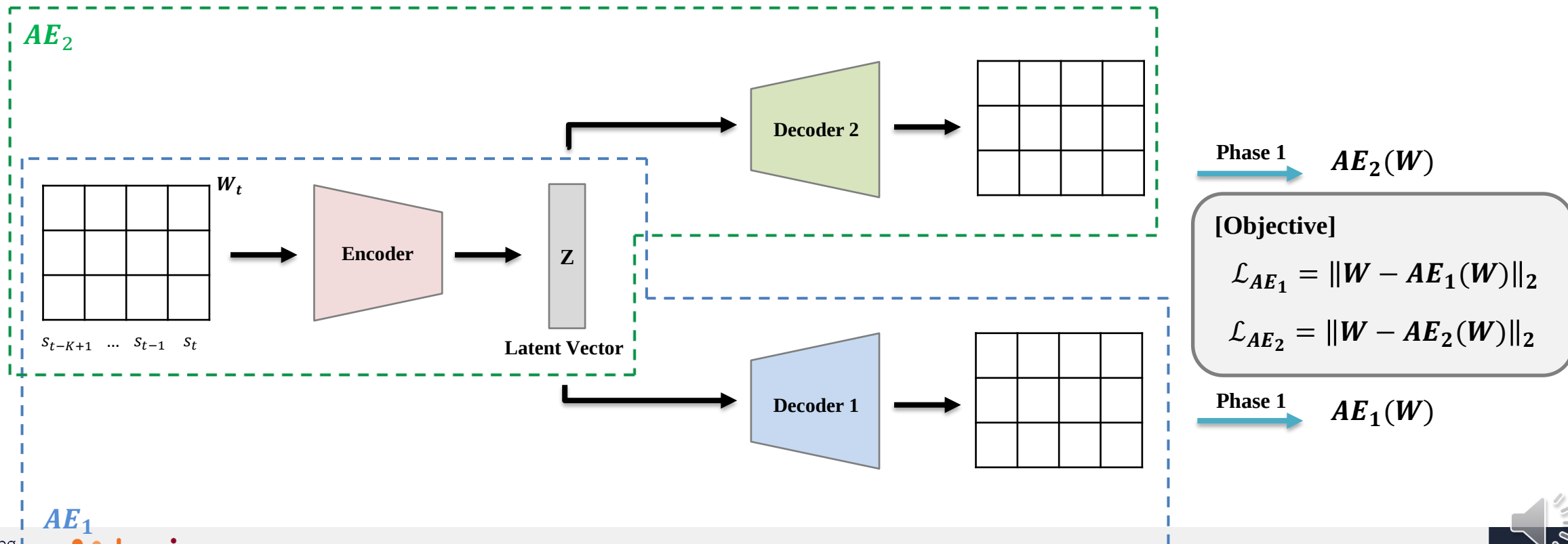
- Autoencoder 한계 극복
 - ✓ Autoencoder 입력 데이터에 비정상 데이터가 포함 여부를 식별하여 우수한 Reconstruction 수행
- GANs 한계 극복
 - ✓ Autoencoder 아키텍처의 도움으로 Adversarial Training의 안정성을 확보 → mode collapse, non-convergence 문제 해결

Autoencoder within An Adversarial Training Framework

Method

❖ Phase1: Autoencoder Training

- 1개의 Encoder와 2개의 Decoder로 구성
 - ✓ AE_1 : Encoder + Decoder 1
 - ✓ AE_2 : Encoder + Decoder 2

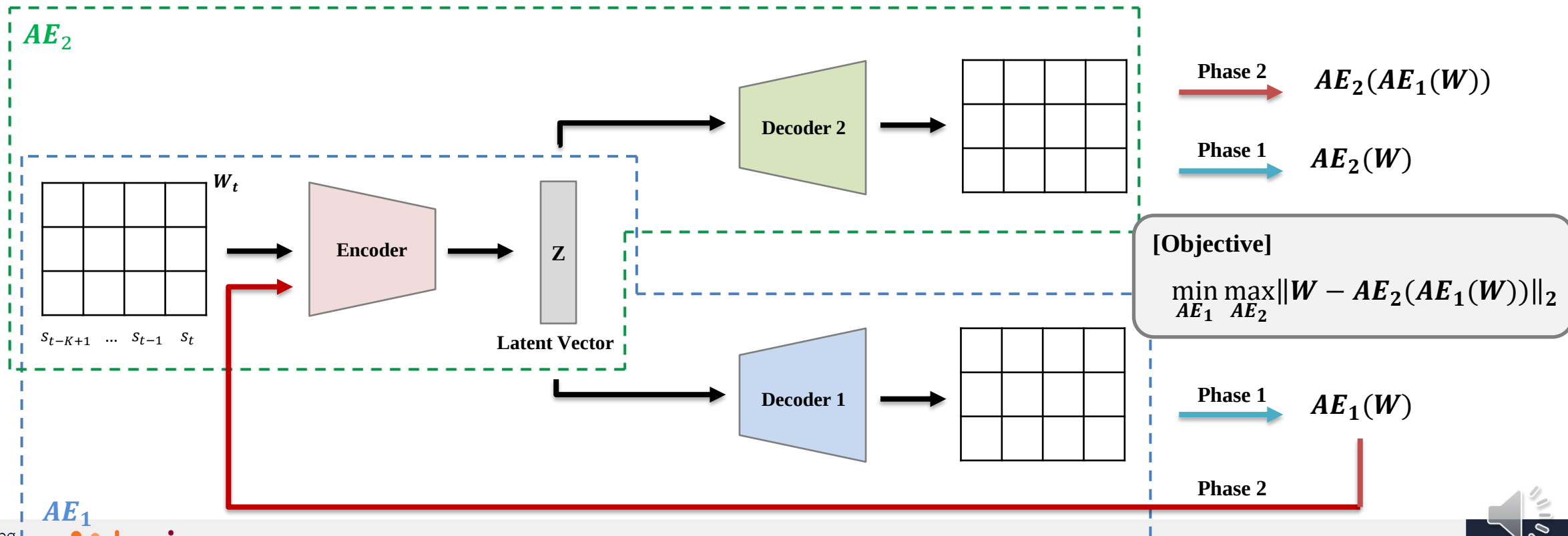


Autoencoder within An Adversarial Training Framework

Method

❖ Phase2: Adversarial Training

- AE_1 을 training 하여 AE_2 를 속일 수 있는 데이터 생성. 즉 실제와 비슷한 데이터를 생성하도록 학습
- AE_2 을 training하기 위해 입력값으로 AE_1 의 output($AE_1(W)$)을 사용 $\rightarrow$ AE_2 가 실제 데이터와 가상 데이터(이상 데이터)인 $AE_1(W)$ 을 구분하도록 학습
- AE_1 는 generator, AE_2 는 discriminator 역할 수행



Autoencoder within An Adversarial Training Framework

Method

❖ Two-phase Training

- Loss Function 최소화 조건
 - ✓ $AE_1 \rightarrow$ generator 역할
 - Phase1: 실제 데이터와 AE_1 의 생성 데이터의 Reconstruction Error 최소화
 - Phase2: 실제 데이터와 가상 데이터를 이용한 AE_2 의 생성 데이터의 Reconstruction Error 최소화
 - ✓ $AE_2 \rightarrow$ discriminator 역할
 - Phase1: 실제 데이터와 AE_2 의 생성 데이터가 Reconstruction Error 최소화
 - Phase2: 실제 데이터와 가상 데이터를 이용한 AE_2 의 생성 데이터가 Reconstruction Error 최대
- n : Training epoch
 - ✓ Training이 반복될 수록 Adversarial training(Phase2)에 가중치

$$\mathcal{L}_{AE_1} = \frac{1}{n} \|W - AE_1(W)\|_2 + \left(1 - \frac{1}{n}\right) \|W - AE_2(AE_1(W))\|_2$$
$$\mathcal{L}_{AE_2} = \frac{1}{n} \|W - AE_2(W)\|_2 - \left(1 - \frac{1}{n}\right) \|W - AE_2(AE_1(W))\|_2$$

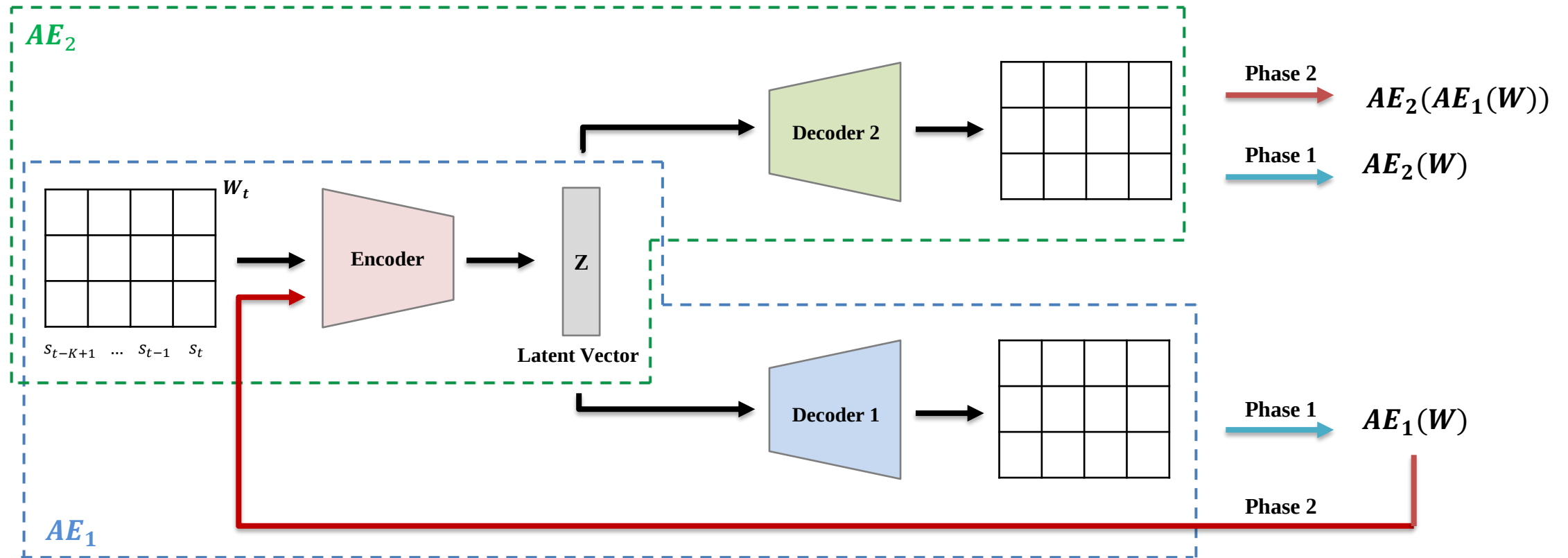
Reconstruction Error of W(Phase 1)

Reconstruction Error of the reconstructed output of AE_2 (Phase 2)

Autoencoder within An Adversarial Training Framework

Method

❖ Two-phase Training



$$\mathcal{L}_{AE_1} = \frac{1}{n} \|W - AE_1(W)\|_2 + (1 - \frac{1}{n}) \|W - AE_2(AE_1(W))\|_2$$

$$\mathcal{L}_{AE_2} = \frac{1}{n} \|W - AE_2(W)\|_2 - (1 - \frac{1}{n}) \|W - AE_2(AE_1(W))\|_2$$

Autoencoder within An Adversarial Training Framework

Method

❖ Inference

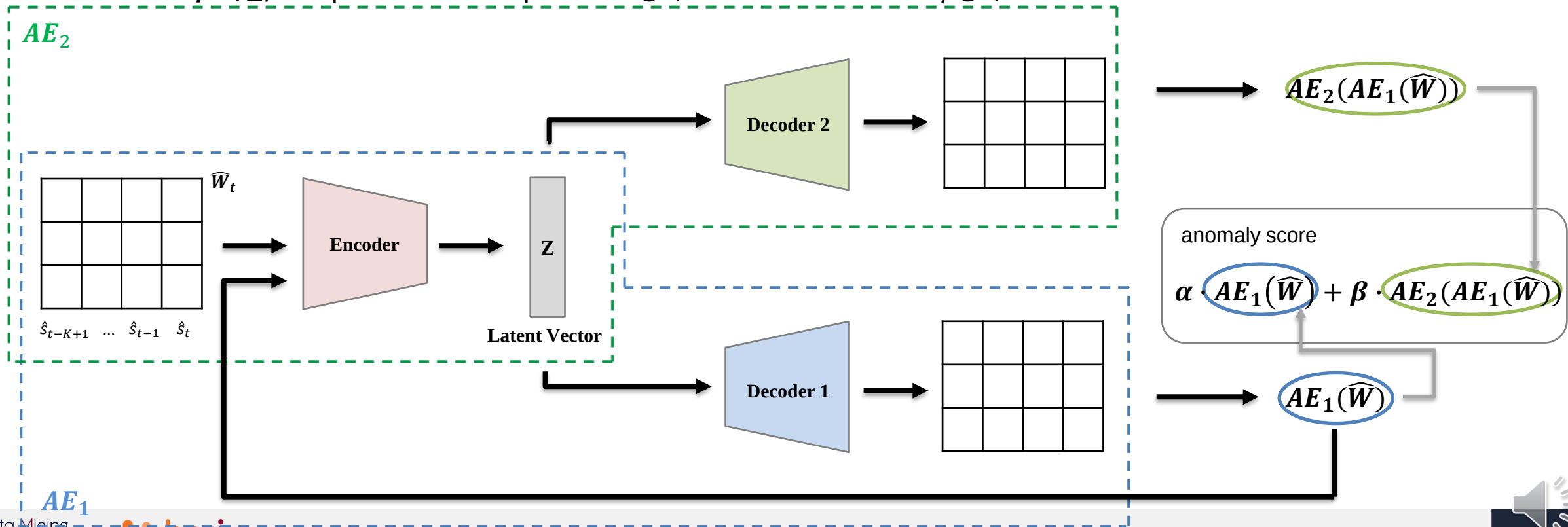
- Anomaly score

$$\alpha \|\widehat{W} - AE_1(\widehat{W})\|_2 + \beta \|\widehat{W} - AE_2(AE_1(\widehat{W}))\|_2$$

- $\alpha + \beta = 1$

✓ $\alpha > \beta$ 이면, true positives & false positives 감소 → detection sensitivity 감소

✓ $\alpha < \beta$ 이면, true positives & false positives 증가 → detection sensitivity 증가



Autoencoder within An Adversarial Training Framework

Experiments and Results

❖ Dataset

- Public Dataset

1. Secure Water Treatment (SWaT)
2. Water Distribution (WADI)
3. Server Machine Dataset(SMD)
4. Soil Moisture Active Passive(SMAP) satellite
5. Mars Science Laboratory (MSL) rover Datasets

Dataset	Train	Test	Dimension	Anomalies(%)
SWaT	496800	449919	51	11.98
WADI	1048571	172801	123	5.99
SMD	708405	708420	28*38	4.16
SMAP	135183	427617	55*25	13.13
MSL	58317	73729	27*55	10.72

Autoencoder within An Adversarial Training Framework

Experiments and Results

❖ Overall performance (1)

- Evaluation Metrics

- ✓ $F1$ score($F1$)

- ✓ $F1^*$ score($F1^*$): Average precision과 Average recall로 측정된 $F1$ score

- Existing Solutions

① Isolation Forests(IF) ② autuoencoder(AE)

③ LSTM-VAE ④ DAGMM ⑤ OmniAnomaly ⑥USAD

- Point-adjust

- ✓ Window 내에서 한 시점이라도 비정상이라면, 해당 구간을 비정상으로 판정

- ✓ SWaT와 WADI 에서만 확인: 우측 Table에서 Without / With로 표기

- ✓ 나머지 데이터셋에는 적용되지 않음

Methods	SWaT						WADI					
	Without			With			Without			With		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
AE	0.9903	0.6295	0.7697	0.9913	0.7040	0.8233	0.9947	0.1310	0.2315	0.3970	0.3220	0.3556
IF	0.9512	0.5884	0.7271	0.9620	0.7315	0.8311	0.2992	0.1583	0.2071	0.6241	0.6155	0.6198
LSTM-VAE	0.9897	0.6377	0.7756	0.7123	0.9258	0.8051	0.9947	0.1282	0.2271	0.4632	0.3220	0.3799
DAGMM	0.4695	0.6659	0.5507	0.8292	0.7674	0.7971	0.0651	0.9131	0.1216	0.2228	0.1976	0.2094
OmniAnomaly	0.9825	0.6497	0.7822	0.7223	0.9832	0.8328	0.9947	0.1298	0.2296	0.2652	0.9799	0.4174
USAD	0.9851	0.6618	0.7917	0.9870	0.7402	0.8460	0.9947	0.1318	0.2328	0.6451	0.3220	0.4296

Methods	SMD				SMAP				MSL			
	P	R	F1	F1*	P	R	F1	F1*	P	R	F1	F1*
AE	0.8825	0.8037	0.8280	0.8413	0.7216	0.9795	0.7776	0.8310	0.8535	0.9748	0.8792	0.9101
IF	0.5938	0.8532	0.5866	0.7003	0.4423	0.5105	0.4671	0.4739	0.5681	0.6740	0.5984	0.6166
LSTM-VAE	0.8698	0.7879	0.8083	0.8268	0.7164	0.9875	0.7555	0.8304	0.8599	0.9756	0.8537	0.9141
DAGMM	0.6730	0.8450	0.7231	0.7493	0.6334	0.9984	0.7124	0.7751	0.7562	0.9803	0.8112	0.8537
OmniAnomaly	0.9809	0.9438	0.9441	0.9620	0.7585	0.9756	0.8054	0.8535	0.9140	0.8891	0.8952	0.9014
USAD	0.9314	0.9617	0.9382	0.9463	0.7697	0.9831	0.8186	0.8634	0.8810	0.9786	0.9109	0.9272

Autoencoder within An Adversarial Training Framework

Experiments and Results

❖ Overall performance (2)

- 전체적으로 제안 모델인 USAD의 성능이 좋음
- Isolate Forest(IF) / DAGMM
 - ✓ 시간 정보(Temporal Information)을 반영하지 않는 모델
 - ✓ 성능이 낮음
- Autoencoder(AE) / LSTM-VAE
 - ✓ 시간 정보(Temporal Information)을 반영하는 모델
 - ✓ 정상 데이터에 가까운 비정상 데이터 탐지 어려움
- OmniAnomaly(OA) / USAD
 - ✓ Autoencoder 기반 방법의 단점을 보완

표.Point-adjust를 적용한 모든 데이터셋의 평균 성능(±표준편차)

	Precision	Recall	<i>F1</i>	<i>F1</i> *
IF	0.64(0.17)	0.68(0.11)	0.62(0.12)	0.60(0.09)
DAGMM	0.62(0.21)	0.76(0.29)	0.65(0.22)	0.79(0.04)
AE	0.77(0.21)	0.76(0.24)	0.73(0.19)	0.86(0.04)
LSTM-VAE	0.72(0.15)	0.80(0.25)	0.75(0.18)	0.86(0.04)
OA	0.73(0.25)	0.95(0.04)	0.78(0.19)	0.91(0.04)
USAD	0.84(0.12)	0.80(0.25)	0.79(0.18)	0.91(0.04)

Autoencoder within An Adversarial Training Framework

Experiments and Results

❖ Effect of parameters

- Sensitivity Threshold
 - ✓ Anomaly score

$$\alpha \|\widehat{W} - AE_1(\widehat{W})\|_2 + \beta \|\widehat{W} - AE_2(AE_1(\widehat{W}))\|_2$$

- $\alpha + \beta = 1$
 - α : AE1 autoencoder의 reconstruction에 가중치
 - β : AE2 autoencoder의 reconstruction에 가중치
- ✓ α 증가 & β 감소
- True Positive(TP)의 감소에 비해 False Positive(FP)가 더 많이 감소함
- ✓ 생산 환경에 적절한 α, β 셋팅 필요

표. Sensitivity Thresholds에 따른 이상탐지 결과(SWaT 데이터 셋)

α	β	FP	TP	F1
0.0	1.0	604	35,616	0.7875
0.1	0.9	580	35,529	0.7853
0.2	0.8	571	35,285	0.7833
0.5	0.5	548	34,590	0.7741
0.7	0.3	506	34,548	0.7738
0.9	0.1	299	34,028	0.7684

50%
감소

3%
감소

Autoencoder within An Adversarial Training Framework

Experiments and Results

❖ Ablation Study

- Two-Phase Training 영향
 - ✓ SMD, SMAP, MSL Dataset 사용
 - ✓ 한 개의 phase를 사용하는 것 보다 두 개의 phase를 같이 사용하는 성능이 더 좋음
 - ✓ Phase 영향 분석
 - Combined > Autoencoders
 - Adversarial train 영향
 - Adversarial train을 도입함으로써 reconstruction error가 증폭
 - Adversarial train 이 적용되지 않으면, 정상데이터와 유사한 비정상데이터를 탐지하기가 어려움
 - Combined > Adversarial
 - Autoencoders 영향
 - Autoencoder 단계가 모델의 안정화 영향을 미침

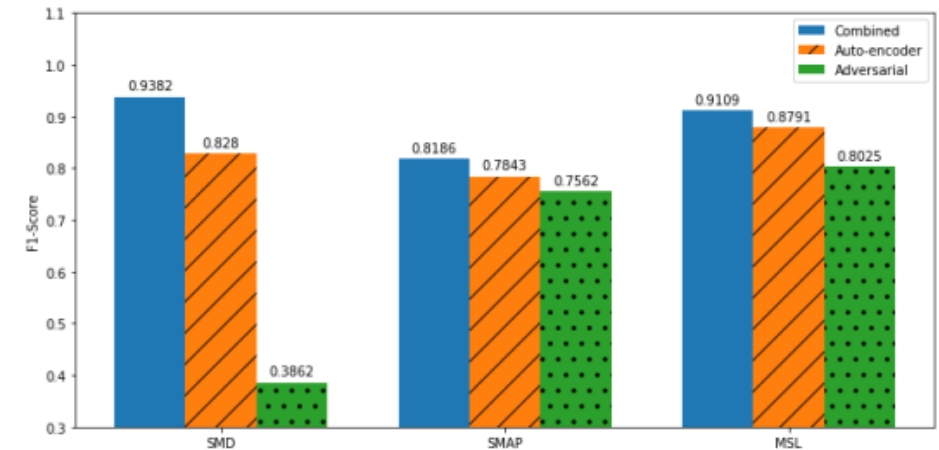


그림. Adversarial training 유무에 따른 영향도

- Combined : USAD
- Autoencoders : USAD phase 1 Training
- Adversarial : USAD phase 2 Training

목차

I. Introduction

- Anomaly Detection for Time Series

II. Autoencoder

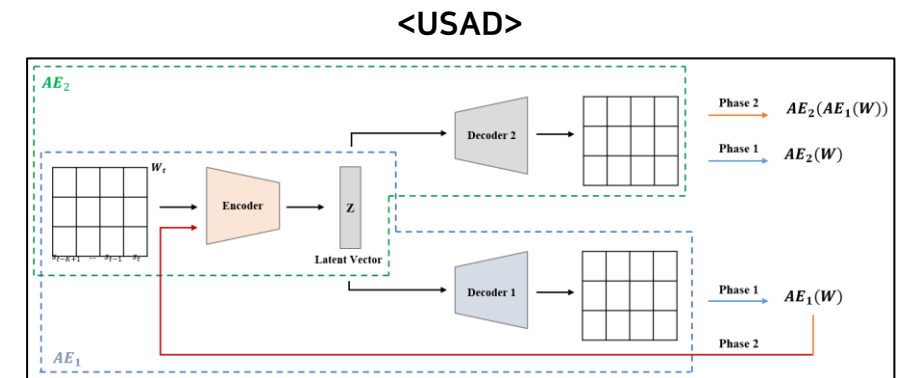
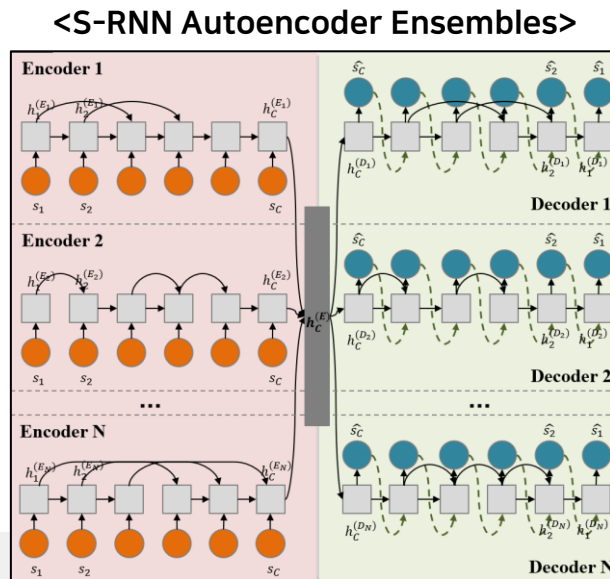
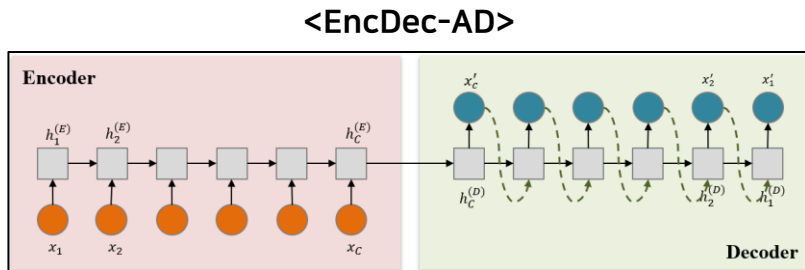
III. Anomaly Detection for Time Series with Autoencoder

- LSTM-Autoencoder
- Autoencoder Ensembles
- Autoencoder within An Adversarial Training Framework

IV. Conclusion

Conclusion

- 시계열 데이터의 이상 탐지 문제는 **도메인 의존성, 다양한 이상 유형, 라벨링 데이터 부족, 데이터 불균형** 등으로 어려움
- 비지도 학습 모델인 Autoencoder가 이상 탐지 문제를 해결하는 도움이 됨
- 하지만 Autoencoder가 갖고 있는 **과적합, 정상데이터와 유사한 비정상데이터 판별** 등 문제점이 있음
- Autoencoder를 기반으로한 시계열 데이터의 이상탐지 문제들은 **Autoencoder의 문제점을 해결하기 위한 방향**으로 발전
- 이 밖에도 굉장히 다양한 모델이 있고, 산업 현장의 문제에 따라 적합한 모델 적용이 필요함



감사합니다

