

2024 겨울 세미나

Representation learning for time series



Sogang University

Vision & Display Systems Lab, Dept. of Electronic Engineering



Presented By

박예인

Outline

- Background
- “Ti-MAE: Self-Supervised Masked Time Series Autoencoders”, Under review on International Conference on Learning Representations (ICLR), 2023.
- “Siamese Network-Based Health Representation Learning and Robust Reference-Based Remaining Useful Life Prediction”, IEEE Transactions on Industrial Informatics, 2022.

Background

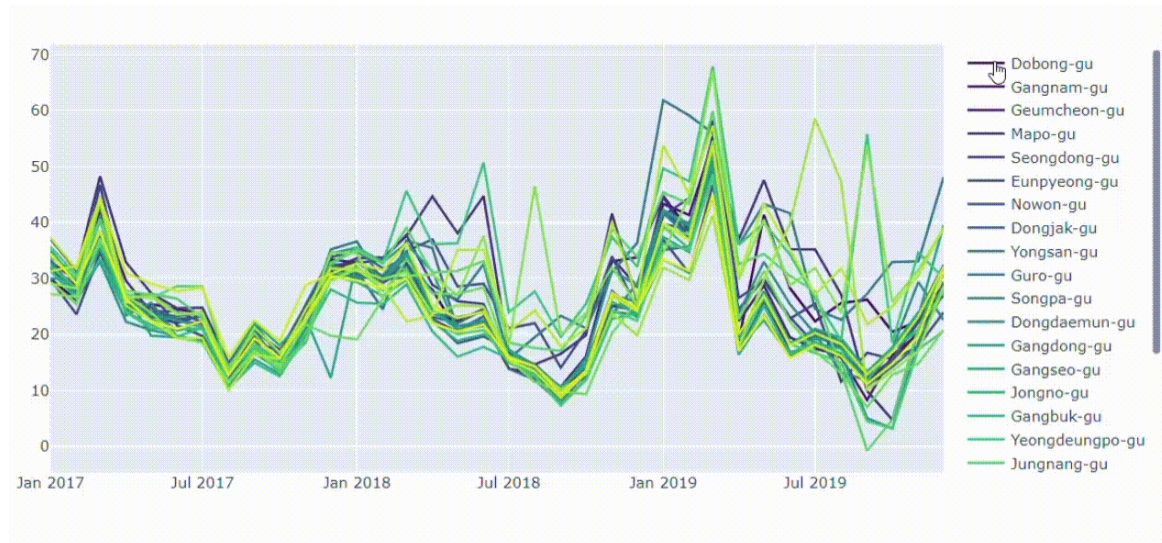
- Time series domain

- 일정 기간 동안 수집된 일련의 순차적인 데이터

- 시간 측면에서 순서가 매겨져 있음
 - 연속적인 관측치는 서로 상관관계가 있음

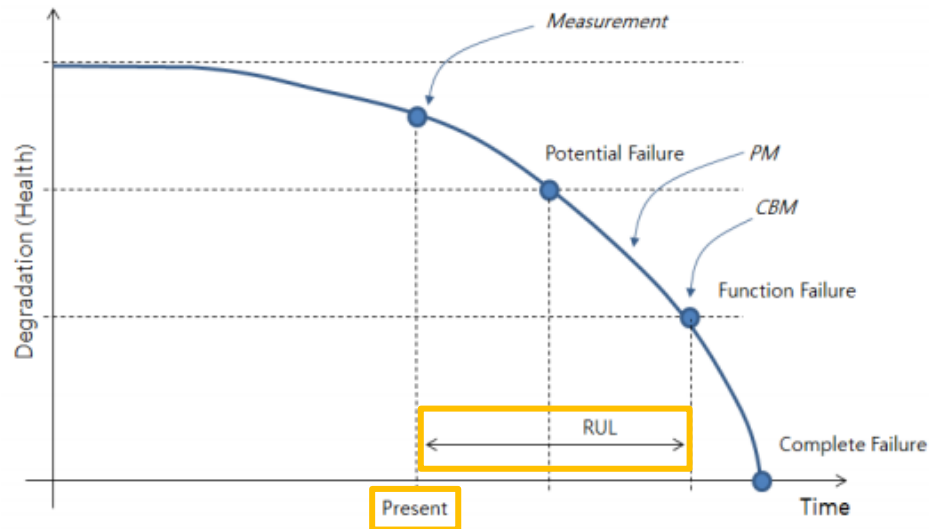
- 시계열 데이터 분석의 목적

- 과거 시점의 시계열 데이터가 갖고 있는 규칙성을 발견해 이를 모형화 / 모델링하는 것
 - 추정된 모형 / 모델을 통해 미래 시점을 예측하는 것



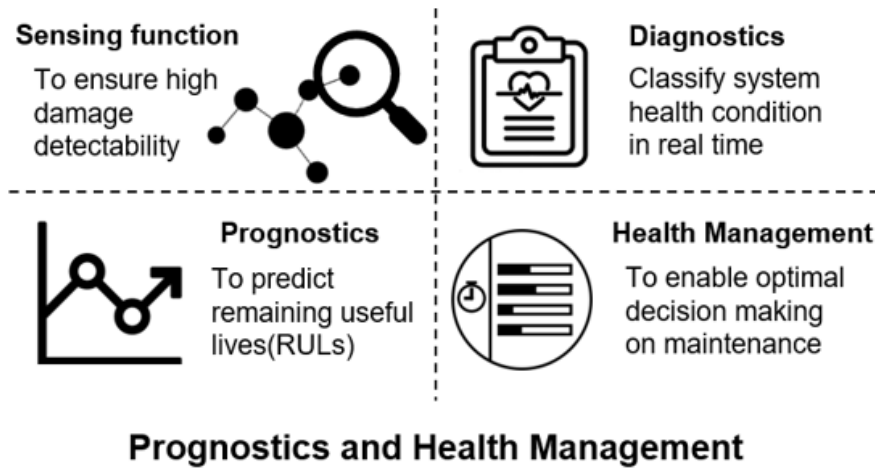
Background

- Necessity of research on remaining useful life prediction
 - 딥러닝 기반의 잔여 유효 수명예측 시스템
 - AI가 예측하는 수명 곡선을 통하여 잔여수명을 예측하고 기계의 안정성 유지
 - 산업 현장에서 사용되는 기계 주요 부품들의 성능 저하로 인한 문제 방지
 - 항공기 산업 및 발전소 등 다양한 분야에서 활용됨
 - 기계가 의도한 기능을 수행할 수 있는 정도



Background

- Necessity of research on remaining useful life prediction
 - PHM (prognostic health management, 예후 상태 관리) 기술의 중요성 증가
 - 시스템 작동을 지속적으로 모니터링하고 장애 수준이나 사용할 수 없는 조건이 발생할 때 비정상을 진단
 - 유지관리 효율성, 안정 작동성, 시스템 성능 개선 면에서 중요성 증가
 - 해당 기술을 통해 필요한 경우에만 condition-based maintenance 수행할 수 있으며 유지 관리 비용을 크게 줄일 수 있음
 - 이때, prognostics는 remaining useful life (RUL, 잔여 수명)를 예측



Background

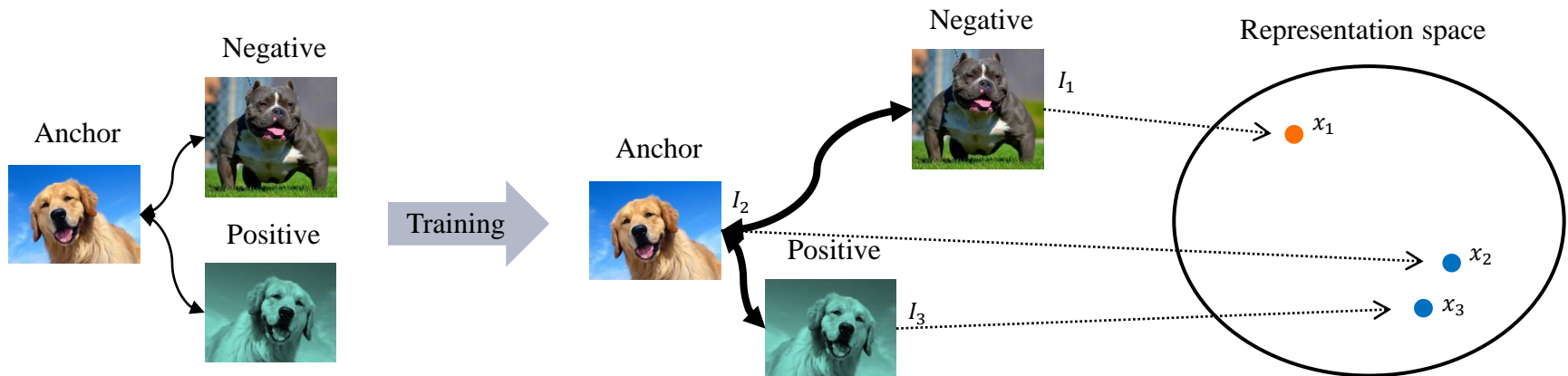
- Contrastive learning

- Representation learning의 한 방법

- Input sample 간의 비교를 통한 representation learning

- “유사한” 데이터는 가깝고 “다른” 데이터는 멀도록 representation space를 학습함

- Input sample 간의 similarity는 label을 통해 학습됨



Background

- Contrastive learning

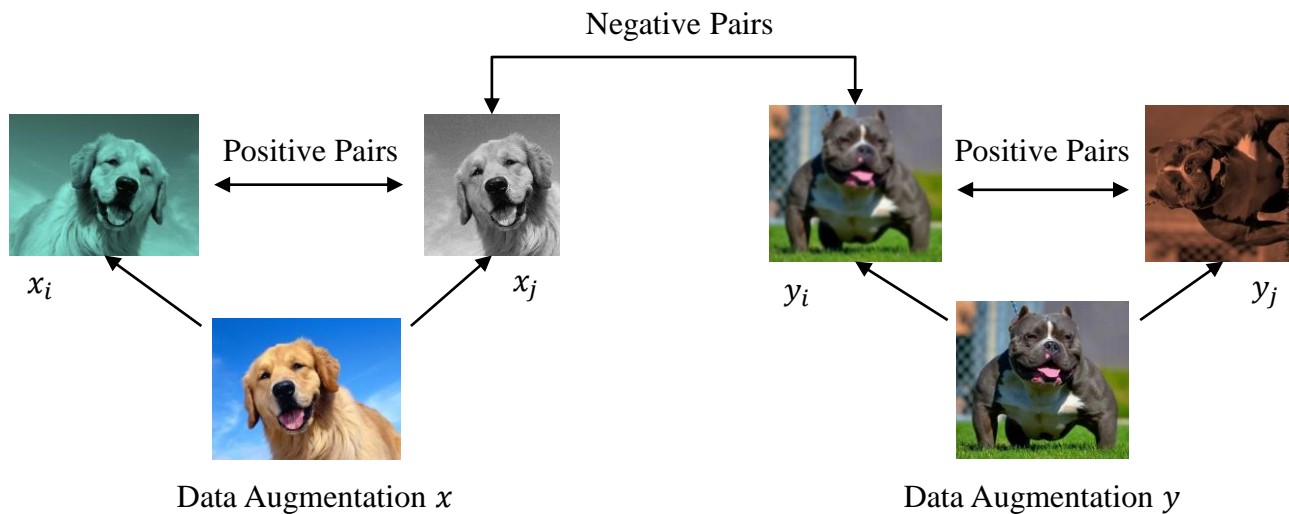
- Data augmentation을 통한 input pair generation

- Image domain에서의 data augmentation

- ※ Positive pair → 동일 이미지상

- ※ Negative pair → 다른 이미지상

- ※ Random crop, rotation, resizing, shifting, noising, blur, color distortion, perspective distortion



Background

- Contrastive learning

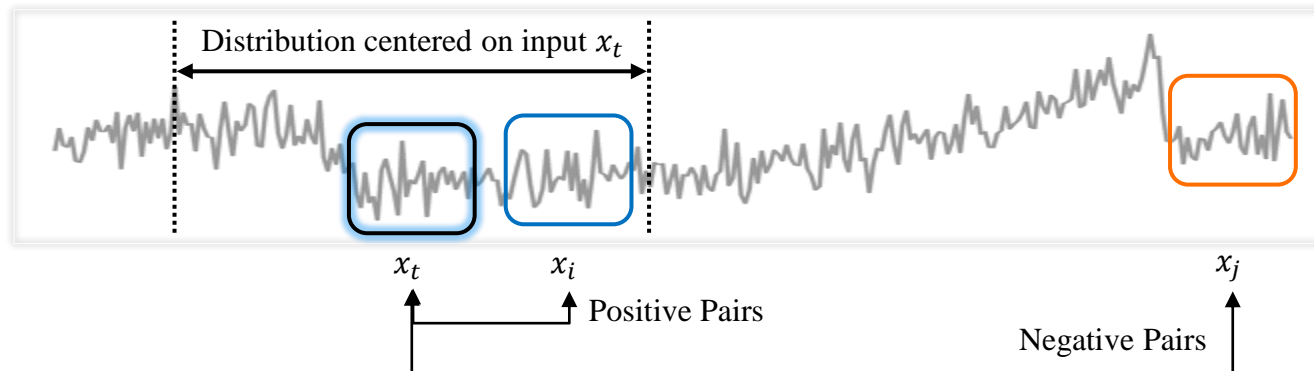
- Data augmentation을 통한 input pair generation

- 시계열 도메인에서의 data augmentation

- ※ Input-centered distribution

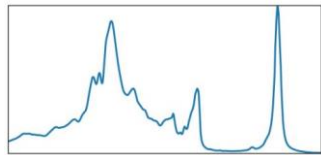
- ✓ Positive pair → 입력 중심 분산 또는 이웃에 속함

- ✓ Negative pair → 입력 중심 분산 또는 이웃에 속하지 않음

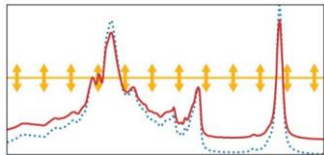


Background

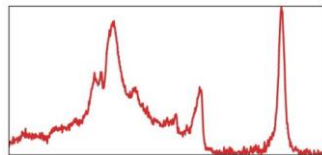
- Contrastive learning
 - Data augmentation을 통한 input pair generation
 - 시계열 도메인에서의 data augmentation
 - ※ Scaling, jittering, window slicing, time warping
 - ※ Random masking
 - ※ Random warping
 - ※ Random reordering



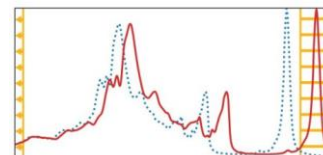
Original



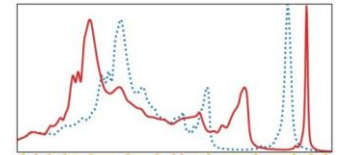
Scaling



Jittering



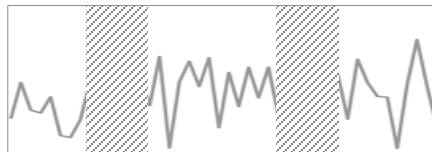
Window slicing



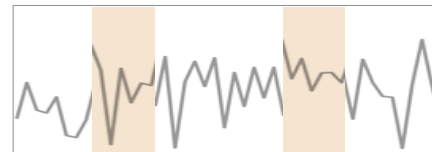
Time warping



Original



Random masking



Random warping



Random reordering

“Ti-MAE: Self-Supervised Masked Time Series Autoencoders”, Under review on International Conference on Learning Representations (ICLR), 2023.

Abstract

- 기존 시계열 데이터 기반 연구에서 contrastive learning 기반의 self-supervised representation learning 방법이 많이 적용됨
 - 입력 샘플간 비교를 통한 contrastive learning
 - Classification 태스크에서 좋은 성능을 보이거나, forecasting에서는 다른 supervised 방법 대비 확연한 성능 격차가 존재함
- 전체 분포를 고려한 random masking 기반의 Ti-MAE라는 새로운 프레임워크 제안
 - 입력 시퀀스를 random masking하고 autoencoder를 통해 point-level에서 재구성함
 - Mask modeling을 채택하고 원본 데이터의 활용도를 유지함
- Masked autoencoding 프레임워크를 통해 raw data에서 강력한 representation을 학습하여 시계열 forecasting 및 classification 작업에서 높은 성능을 보임

Proposed method

- Transformer 기반의 Ti-MAE

- 입력 시계열 데이터의 일부분을 random masking 처리

- 다양한 서로 다른 토큰들의 조합으로 모든 입력 시퀀스를 활용함

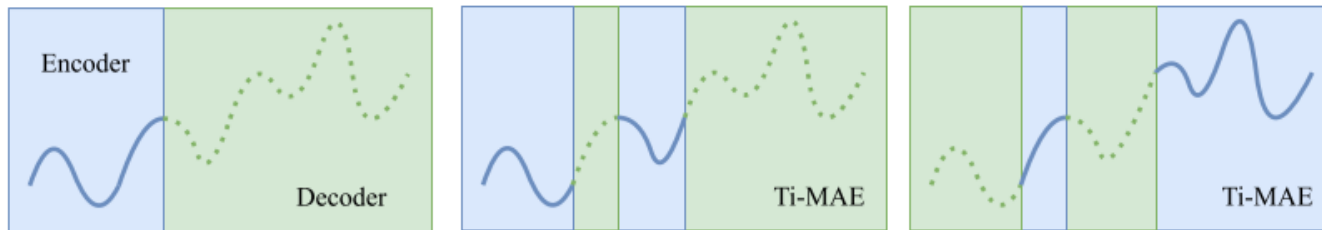
- ※ Fixed continuous masking은 전체

- Random masking은 입력의 전체 분포를 고려하기 때문에 데이터 분포 이동 문제 완화 가능

- Masking 비율의 유연한 설정으로 다양한 크기의 여러 time window에 대해 예측 가능

- Autoencoder를 통해 마스킹된 부분을 point-level에서 재구성

- 데이터의 전반적인 문맥 학습 가능



(a) End-to-end forecasting.

(b) Random masking applied in Ti-MAE.

미래 시계열만 마스킹하고 재구성함

Random masking을 통해 전체 입력 시계열을 완전히 활용하여
각 iteration마다 encoder에 다양한 입력을 생성함

< 시계열 데이터에 대한 generative transformer 기반 모델의 masking 전략 >

Proposed method

- Transformer 기반의 Ti-MAE

- Input embedding

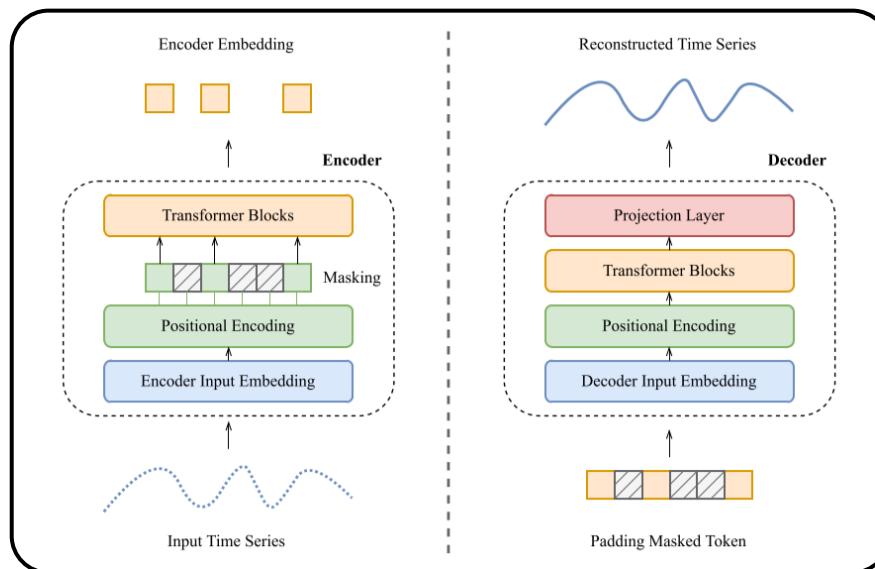
- 하나의 1D convolutional layer만을 사용하여 feature 추출
 - Positional encoding으로 위치 정보 저장

- Masking

- Original temporal data를 timestamp에 따라 토큰화 진행
 - 균일한 분포를 따르며 random sampling 및 나머지 부분 masking

Masking ratio

- 데이터의 정보 밀도 및 중복성과 관련이 있으며 성능에 상당한 영향을 줌
- 자연어의 경우 단어 분포가 뚜렷하여 정보 밀도가 높음
- 이미지의 경우 공간 중복성이 높아 정보 밀도가 낮음
- 시계열 데이터의 경우 local continuity를 가지므로 높은 마스킹 비율을 적용해야 함



< Ti-MAE 구조 >

Experimental results

- 성능 비교

Ratio	MSE	MAE	Case	MSE	MAE	Length	MSE	MAE	Masking Strategy					
0.45	0.3082	0.3557	None	0.2103	0.2696	200	0.2413	0.2844	Random	Continuous	Split	Periodic		
0.60	0.2650	0.3414	Scaling	0.2383	0.3022	300	0.2103	0.2696	MSE	0.2103	0.3834	0.3564	0.2720	
0.75	0.2103	0.2696	Shifting	0.2399	0.3388	400	0.2877	0.3501	MAE	0.2696	0.4420	0.3936	0.3357	
0.90	0.2483	0.3176	Jittering	0.2508	0.3316	500	0.2328	0.2985						

< Masking ratio >

< Data augmentation >

< Input sequence length >

< Masking strategy >

Method	Ti-MAE		CoST		TS2Vec		TNC		MoCo		
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	
ETTh	12	0.2629	0.3462	0.3374	0.4001	0.5817	0.5217	0.6056	0.5389	0.6419	0.5518
	24	0.3520	0.3924	0.3832	0.4301	0.5897	0.5312	0.6331	0.5616	0.6491	0.5630
	48	0.3977	0.4173	0.4342	0.4665	0.6242	0.5545	0.6934	0.6001	0.6804	0.5867
	96	0.4266	0.4301	0.5229	0.5201	0.6812	0.5699	0.7538	0.6391	0.7618	0.6323
	128	0.4493	0.4436	0.5742	0.5529	0.7190	0.5919	0.7949	0.6622	0.8062	0.6577
	168	0.5091	0.4594	0.6326	0.5838	0.7621	0.6387	0.8360	0.6849	0.8201	0.6742
Weather	12	0.0932	0.1460	0.1652	0.2630	0.1481	0.2367	0.1819	0.2508	0.1642	0.2521
	24	0.1226	0.1809	0.2719	0.3525	0.3011	0.3551	0.3118	0.3731	0.3112	0.3651
	48	0.1633	0.2280	0.3662	0.3672	0.3741	0.4178	0.3803	0.4117	0.3717	0.4163
	96	0.2123	0.2735	0.4119	0.4266	0.4289	0.4507	0.4176	0.4174	0.4077	0.4419
	128	0.2197	0.2805	0.4302	0.4686	0.4663	0.4839	0.4569	0.4824	0.4582	0.4693
	168	0.2460	0.3049	0.4636	0.4914	0.4909	0.5061	0.4789	0.4950	0.4820	0.4992
Exchange	24	0.0697	0.1889	0.1365	0.2721	0.0873	0.2245	0.0834	0.2084	0.1058	0.2553
	48	0.1255	0.2448	0.2532	0.3783	0.1666	0.3047	0.1648	0.2928	0.2018	0.3588
	96	0.1701	0.2972	0.5408	0.5645	0.4686	0.5098	0.3756	0.4510	0.4162	0.5002
	128	0.2208	0.3242	0.6786	0.6334	0.6540	0.6036	0.5483	0.5441	0.5950	0.6050
	168	0.2151	0.3316	0.8859	0.7338	0.9683	0.7348	0.7701	0.6470	0.8079	0.6997
	196	0.2123	0.3291	0.9720	0.7703	1.1692	0.8084	0.9495	0.7204	0.9534	0.7591
ILI	24	2.7474	1.0740	2.8332	1.0656	3.5111	1.1882	3.3729	1.2011	2.9399	1.1014
	36	2.7124	1.0348	3.1439	1.1197	3.7813	1.2588	4.0722	1.3292	3.4974	1.2212
	48	2.6138	1.0399	3.4153	1.1725	4.1892	1.3319	4.1239	1.3239	3.7872	1.2713
	60	2.2889	0.8940	3.7917	1.2553	4.2588	1.3352	3.9937	1.3063	3.8137	1.2695
	72	2.0820	0.8372	4.0823	1.3232	4.1868	1.3431	4.0423	1.3294	3.8818	1.3023
	96	2.4419	1.0287	4.2442	1.3755	4.3677	1.3756	4.2162	1.3594	4.2148	1.3530

Method	Ti-MAE†		ETSformer		FEDformer		Autoformer		Informer		
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	
ETTh	12	0.2826	0.3383	0.4479	0.4582	0.3272	0.3940	0.5016	0.5204	0.4299	0.4644
	24	0.3430	0.3816	0.4602	0.4621	0.3699	0.4185	0.5063	0.5309	0.4880	0.4963
	48	0.3705	0.3939	0.4855	0.4735	0.3912	0.4347	0.5703	0.5563	0.6625	0.5774
	96	0.4039	0.4074	0.5090	0.4851	0.4194	0.4476	0.6052	0.5663	0.9584	0.7157
	128	0.4270	0.4208	0.5279	0.4949	0.4360	0.4551	0.6043	0.5726	0.9504	0.7197
	168	0.4455	0.4363	0.5446	0.5044	0.4733	0.4783	0.7382	0.6199	1.1043	0.7867
Weather	12	0.0811	0.1199	0.0900	0.1537	0.1476	0.2350	0.2042	0.2960	0.2351	0.3128
	24	0.1065	0.1484	0.1396	0.2224	0.1624	0.2496	0.2200	0.3141	0.1244	0.2022
	48	0.1290	0.1784	0.1848	0.2735	0.1993	0.2898	0.2691	0.3542	0.2352	0.3129
	96	0.1633	0.2151	0.2034	0.2994	0.2350	0.3139	0.2891	0.3673	0.2808	0.3586
	128	0.1774	0.2283	0.2092	0.2972	0.2395	0.3148	0.2758	0.3469	0.3055	0.3723
	168	0.2031	0.2525	0.2199	0.3016	0.2632	0.3281	0.2861	0.3506	0.3473	0.4003
Exchange	24	0.0276	0.1167	0.0266	0.1130	0.0717	0.1958	0.0894	0.2239	0.4963	0.5623
	48	0.0438	0.1481	0.0441	0.1464	0.0954	0.2247	0.1474	0.2881	1.0477	0.8169
	96	0.0814	0.2074	0.0861	0.2044	0.1470	0.2790	0.2883	0.3957	1.1038	0.8215
	128	0.1108	0.2361	0.1153	0.2373	0.1886	0.3153	0.3102	0.4107	1.1978	0.8535
	168	0.1443	0.2824	0.1549	0.2773	0.2484	0.3638	0.3066	0.4108	1.1564	0.8444
	196	0.1661	0.3040	0.1830	0.3034	0.2718	0.3800	0.2990	0.4021	1.1679	0.8545
ILI	24	2.4781	0.9925	3.1358	1.2128	3.3017	1.2689	3.3292	1.2088	4.2526	1.3551
	36	2.2103	0.8956	2.9369	1.1218	2.6125	1.0575	3.4076	1.1688	4.7647	1.4433
	48	1.9697	0.8826	2.9386	1.1120	2.5883	1.0683	3.2077	1.1125	4.8189	1.4553
	60	2.3496	0.9545	2.8840	1.1324	2.8460	1.1533	3.3373	1.1659	4.7974	1.4669
	72	2.1563	0.8884	2.8615	1.1579	2.8921	1.1721	3.1079	1.1237	4.1188	1.3718
	96	2.3860	0.9827	3.1109	1.2186	3.1048	1.2412	3.0530	1.1260	4.5218	1.4401

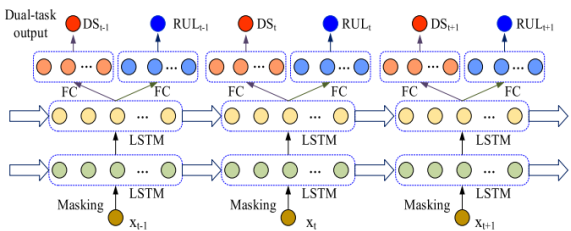
< Representation learning 방법들과 성능 비교 >

< End-to-end 방법들과 성능 비교 >

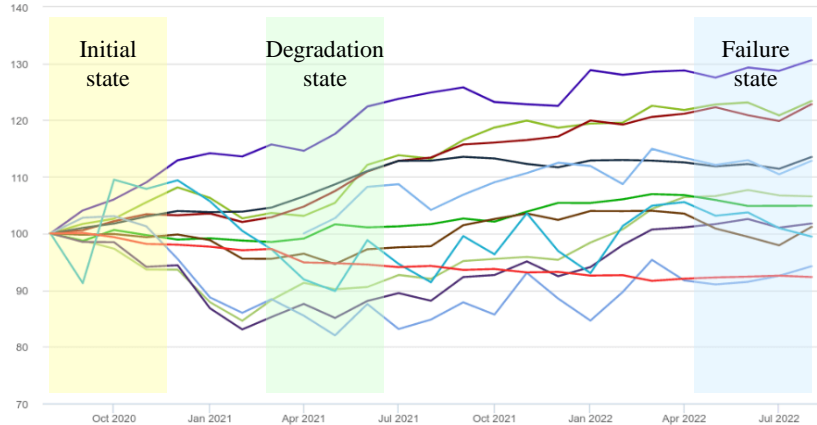
“Siamese Network-Based Health Representation Learning and Robust Reference-Based Remaining Useful Life Prediction”, *IEEE Transactions on Industrial Informatics*, 2022.

Abstract

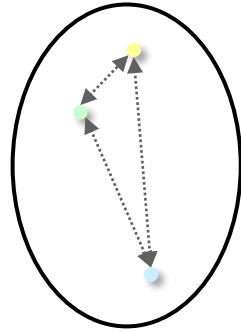
- Siamese network 기반의 health representation learning method를 제안함
 - 부족한 training sample 개수로 인해 deep neural network가 overfitting되는 문제 해결을 위함
- Multitask learning scheme 도입
 - RUL prediction을 학습하는 동시에 sample들에 대한 representation learning 수행
 - Health representation map에서 sample간의 차이가 RUL 값의 차이와 linear relationship을 갖도록 Siamese network 기반 representation learning 제안
 - 학습된 representation space는 degradation을 반영하므로 각 training sample을 test sample의 RUL을 예측하기 위한 reference로 사용 가능
- 제안된 학습 및 예측 방법이 RUL 예측 성능 뿐만 아니라 데이터 부족에 대한 robustness 향상에도 기여함을 실험 결과를 통해 보여줌



< Multitask learning scheme >



Representation space

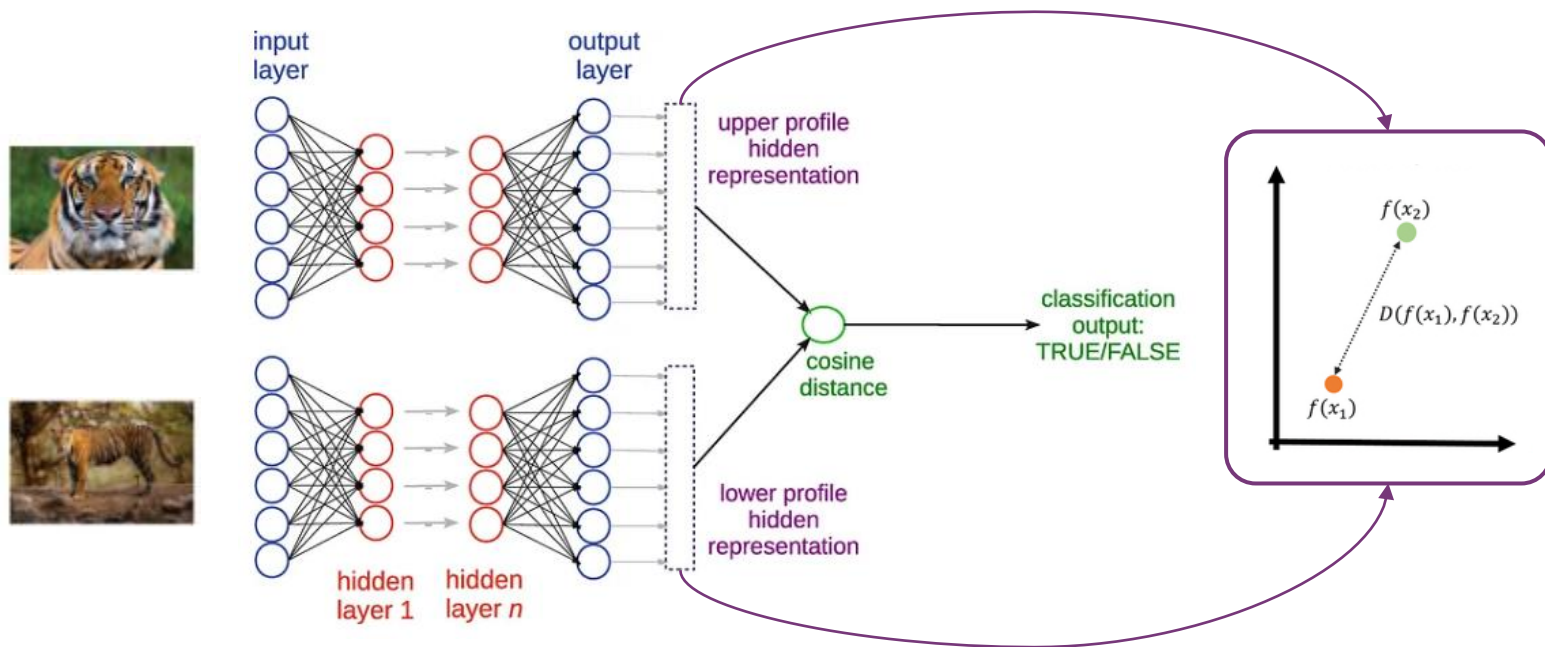


< Representation space에서 데이터 샘플 격차 >

Proposed method

- Siamese network

- Original Siamese network의 경우, 주어진 sample의 class label을 기반으로 embedding space를 학습하도록 제안됨
 - Binary image classification에서 주로 사용되는 네트워크
 - Embedding space에서 동일한 class에 속하는 sample pair는 서로 가까워야 하고, 서로 다른 두 class에 속하는 sample pair는 충분한 거리만큼 떨어져야 함



< Siamese network >

Proposed method

- Siamese network

- Weight sharing 형태의 embedding network F 로 구성됨

- X_i, X_j 는 sample pair를 나타냄

$$\ast X_i = [x_{i1,t:t+TW-1}, x_{i2,t:t+TW-1}, \dots, x_{iK,t:t+TW-1}]$$

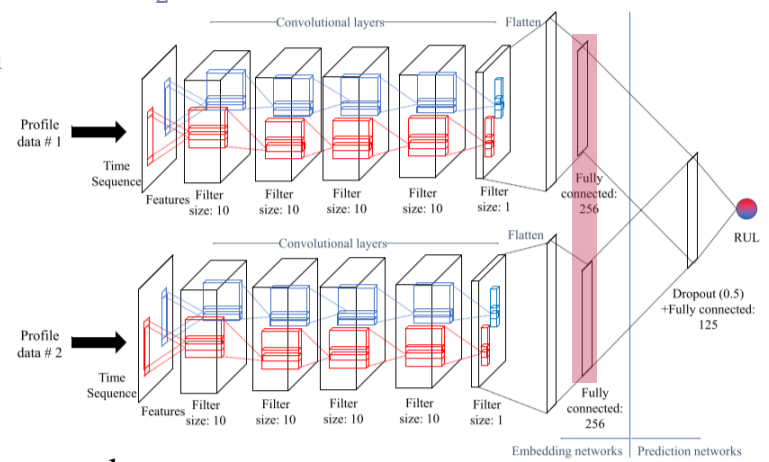
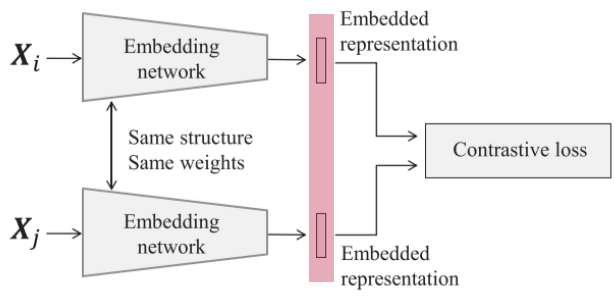
- 각 sample에 대한 embedding representation에 contrastive loss를 적용하여 학습함

$$\text{Contrastive loss} = \frac{1}{2}Y_{ij}d_{ij}^2 + \frac{1}{2}(1 - Y_{ij})\max\{0, \alpha - d_{ij}\}^2$$

- Sample pair (X_i, X_j) 가 동일한 class인 경우, $Y_{ij} = 1, \text{loss} = \frac{1}{2}d^2$

- Sample pair (X_i, X_j) 가 다른 class인 경우, $Y_{ij} = 0, \text{loss} = \frac{1}{2}d^2$

- $d_{ij} = (F(X_i) - F(X_j))$, $\alpha = \text{minimum margin}$



< Proposed Siamese network >

Proposed method

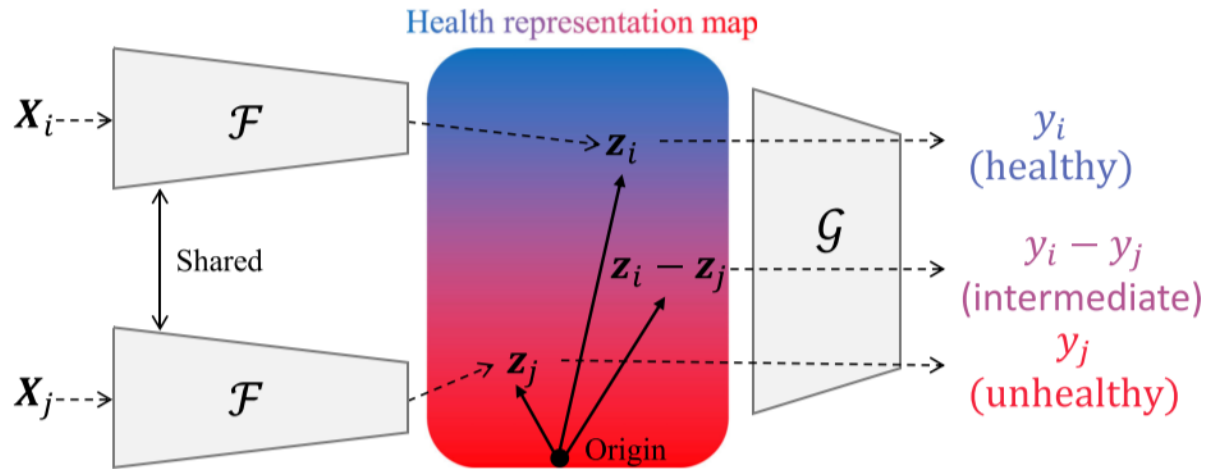
- Siamese network의 embedding space

- Health representation map

- Embedding network F 와 prediction network g 사이 맵
- Multitask learning scheme을 통해 두가지 목적으로 학습됨

※ Health representation map은 RUL을 나타낼 수 있어야 함

- ✓ Embedding network의 output인 z 가 prediction network를 통해 RUL에 잘 매핑되어야 맵에서 health information이 반영될 수 있음



< Health representation map >

Proposed method

- Siamese network의 embedding space

- Health representation map

- 시스템의 degradation dynamics를 캡처해야 함

- ※ RUL 간의 차이가 health representation 간의 차이로 드러나야 함

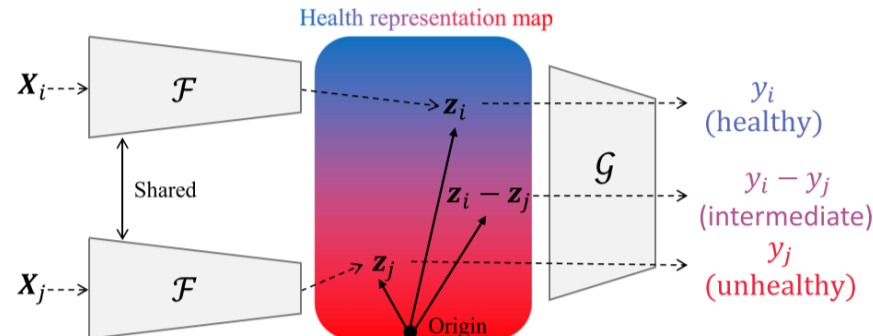
- ※ Difference 기반으로 RUL 간의 차이를 추정하는 학습 방법을 통해 health representation과 RUL 간의 일관성을 유지할 수 있음

- ※ Health representation map에서 샘플이 예상치 못한 position을 가지지 못하도록 함

$\mathcal{G}(z_i - z_j) = y_i - y_j$ 를 만족시키기 위한 학습

$$\min_{\theta_{\mathcal{F}}, \theta_{\mathcal{G}}} \mathcal{L}_{\text{pre}}(\mathbf{X}_i, y_i) = \sqrt{(\mathcal{G}(\mathcal{F}(\mathbf{X}_i)) - y_i)^2}, \quad \min_{\theta_{\mathcal{F}}, \theta_{\mathcal{G}}} \mathcal{L}_{\text{pen}}(\mathbf{X}_i, \mathbf{X}_j, y_i, y_j) = \sqrt{(\mathcal{G}(\mathcal{F}(\mathbf{X}_i) - \mathcal{F}(\mathbf{X}_j)) - (y_i - y_j))^2}$$

$$\min_{\theta_{\mathcal{F}}, \theta_{\mathcal{G}}} \mathcal{L}_{\text{pre}}(\mathbf{X}_i, y_i) + \mathcal{L}_{\text{pre}}(\mathbf{X}_j, y_j) + \lambda \mathcal{L}_{\text{pen}}(\mathbf{X}_i, \mathbf{X}_j, y_i, y_j). \rightarrow \text{Final training objective function}$$



< Health representation map >

Proposed method

- Simple data index sampling 기반 학습
 - 학습에 모든 pair를 고려하는 것은 많은 연산량을 필요로 함

Algorithm 1: Pseudocode for Training.

Require: Embedding network \mathcal{F} , prediction network \mathcal{G} , training dataset \mathcal{D} , batch size N ;

- 1: Compute L , the number of iterations per epoch;
 - 2: Initialize the data index sets Ind_1 and Ind_2 as
 $\text{Ind}_1 = \text{Ind}_2 = \{1, 2, \dots, |\mathcal{D}|\}$, where $|\mathcal{D}|$ is the size of the training dataset;
 - 3: **while** $\theta_{\mathcal{F}}$ and $\theta_{\mathcal{G}}$ have not converged **do**
 - 4: Randomly shuffle Ind_1 and Ind_2 ;
 - 5: **for** $l = 1, 2, \dots, L$ **do**
 - 6: Sample a minibatch $\{\mathbf{X}_{(1)}^{\text{Ind}_1}, \mathbf{X}_{(2)}^{\text{Ind}_1}, \dots, \mathbf{X}_{(N)}^{\text{Ind}_1}\}$ with $\{y_{(1)}^{\text{Ind}_1}, y_{(2)}^{\text{Ind}_1}, \dots, y_{(N)}^{\text{Ind}_1}\}$ according to Ind_1 ;
 - 7: Sample a minibatch $\{\mathbf{X}_{(1)}^{\text{Ind}_2}, \mathbf{X}_{(2)}^{\text{Ind}_2}, \dots, \mathbf{X}_{(N)}^{\text{Ind}_2}\}$ with $\{y_{(1)}^{\text{Ind}_2}, y_{(2)}^{\text{Ind}_2}, \dots, y_{(N)}^{\text{Ind}_2}\}$ according to Ind_2 ;
 - 8: Update $\theta_{\mathcal{F}}$ and $\theta_{\mathcal{G}}$ by applying stochastic gradient descent to
$$\sum_{k=1}^N [\mathcal{L}_{\text{pre}}(\mathbf{X}_{(k)}^{\text{Ind}_1}, y_{(k)}^{\text{Ind}_1}) + \mathcal{L}_{\text{pre}}(\mathbf{X}_{(k)}^{\text{Ind}_2}, y_{(k)}^{\text{Ind}_2}) + \lambda \mathcal{L}_{\text{pen}}(\mathbf{X}_{(k)}^{\text{Ind}_1}, \mathbf{X}_{(k)}^{\text{Ind}_2}, y_{(k)}^{\text{Ind}_1}, y_{(k)}^{\text{Ind}_2})]$$
 - 9: **end for**
 - 10: **end while**
-

Proposed method

- RUL estimation

- Test sample X_k 가 주어졌을 때, 그것의 RUL은 두가지 방법으로 예측됨

- 1) model-based RUL estimation

- ※ Embedding과 prediction network를 통한 direct RUL 예측

$$\checkmark \tilde{y}_k^{mod} = g(F(X_k))$$

- 2) Reference-based RUL estimation

- ※ Health representation map에서 해당 샘플과 다른 샘플간의 차이를 기반으로 RUL 추론

$$\checkmark \tilde{y}_k^{ref} = \frac{1}{|D|} \sum_{(X_j, y_j) \in D} y_j + g(F(X_k) - F(X_j))$$

- ✓ (X_j, y_j) 와 X_k 를 알 때, $y_j + g(F(X_k) - F(X_j))$ 를 계산하여 y_k 추정 가능함

- ✓ $\rightarrow (X_j, y_j)$ 를 y_k 를 추정하기 위한 reference라 함

- ※ 추정치들 결합으로 앙상블 효과를 통해 더 robust한 예측 결과를 제공함

- ✓ 추정치 분산을 줄임

- ✓ 소수의 학습 데이터로 모델이 overfitting될 때 효과가 큼

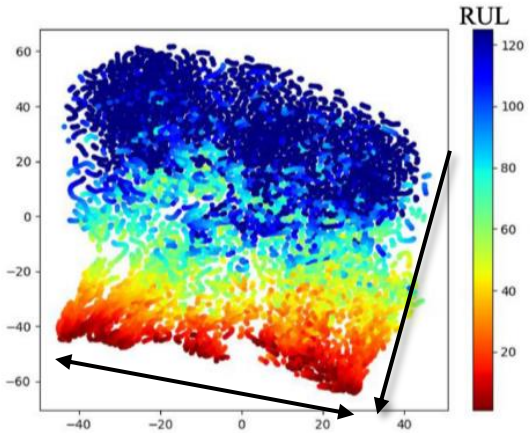
- ✓ 그러나, 각 training sample별로 차이 값 기반 추론을 수행하기 때문에 시간이 많이 소요됨

Experimental results

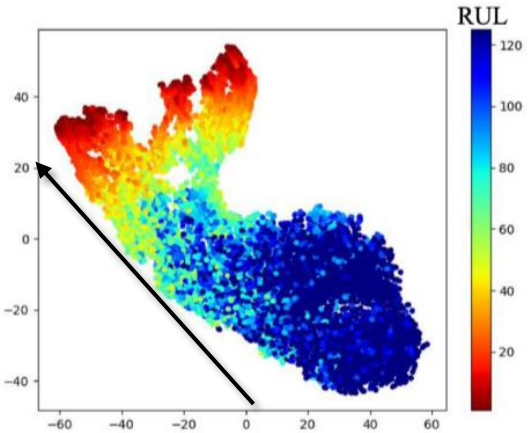
- Health representation map 분석

- Health representation map을 사용한 제안 방법의 경우

- 높은 RUL 샘플에서 낮은 RUL 샘플로의 방향성이 더 명확한 것을 확인할 수 있음



< 기존 CNN >



< 제안 방법 >

Number of centroids	Input	FS-CNN	Health representation map (λ)			
			0.1	0.4	0.7	1.0
3	0.15	0.40	0.42	0.43	0.44	0.39
5	0.14	0.35	0.37	0.39	0.41	0.39
10	0.13	0.33	0.35	0.36	0.35	0.35

Number of centroids	Input	FS-CNN	Health representation map (λ)			
			0.1	0.4	0.7	1.0
3	0.15	0.43	0.44	0.44	0.47	0.46
5	0.11	0.30	0.33	0.35	0.36	0.38
10	0.07	0.24	0.26	0.26	0.22	0.25

< RUL에 따른 샘플 클러스터링 유사성 (NMI) >

0: 상호 정보 없음, 1: 완벽한 상관 관계

< RUL에 따른 샘플 클러스터링 유사성 (ARI) >

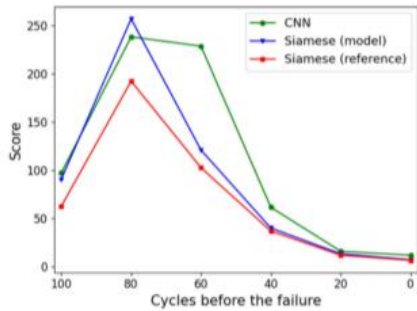
$$\frac{RI - \text{Expected_RI}}{\max(\text{Max_RI} - \text{Expected_RI}, 0)}$$

$$\sum_{y \in Y} \sum_{x \in X} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$

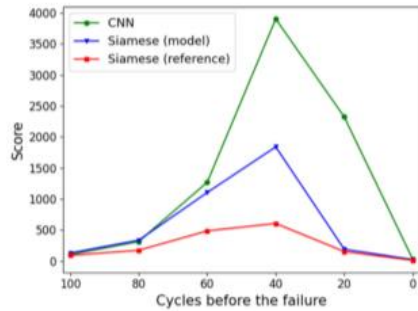
Experimental results

- 제안한 학습 및 예측 방법의 효과성

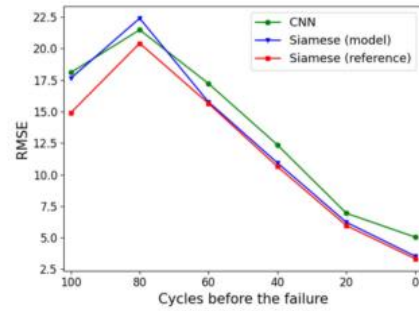
- 기존 CNN, model-based Siamese, reference-based Siamese 네트워크와의 성능 비교



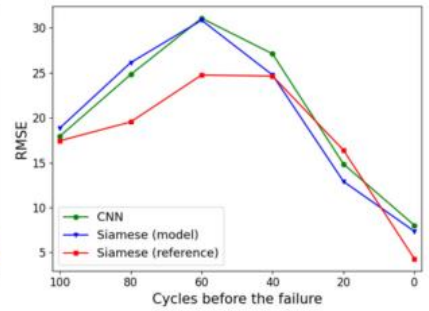
(a)



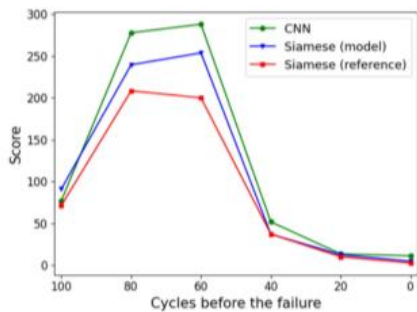
(b)



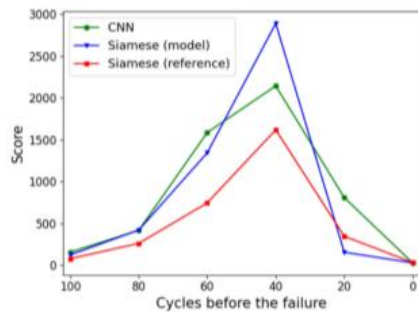
(a)



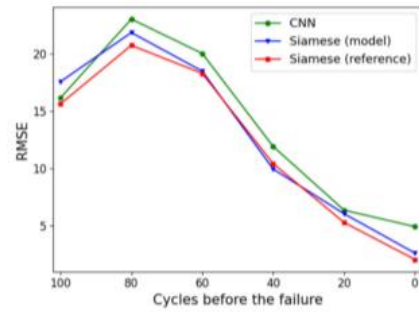
(b)



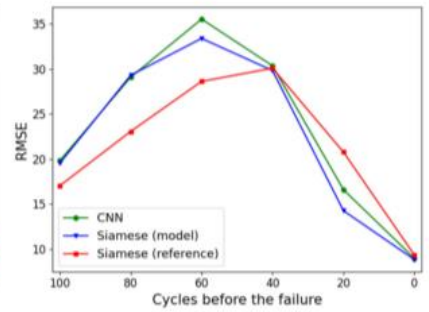
(c)



(d)



(c)



(d)

< Score 성능 비교 >

< RMSE 성능 비교 >

Experimental results

- 제안한 학습 및 예측 방법의 효과성
 - 제한된 학습 데이터 개수에 따른 성능 비교

F D 0 0 1	# of training profiles	10	30	70	All
	LSTM [11]	1,105 (505)	1,878 (3,565)	415 (182)	338
CNN+LSTM [29]	7,822 (19,973)	816 (288)	589 (185)	303	
CNN [10]	1,105 (172)	592 (322)	527 (36)	360	
Siamese (model)	878 (116)	543 (122)	498 (60)	267	
Siamese (reference)	592 (62)	403 (76)	349 (30)	253	
F D 0 0 2	# of training profiles	50	100	200	All
	LSTM [11]	7,265 (4,161)	16,685 (19,890)	14,951 (17,606)	4,450
CNN+LSTM [29]	19,123 (16,283)	27,578 (28,270)	12,300 (6,317)	3,440	
CNN [10]	15,544 (3,314)	6,814 (2,628)	4,460 (2,443)	3,352	
Siamese (model)	15,728 (4,076)	9,448 (3,929)	4,501 (2,424)	2,134	
Siamese (reference)	6,548 (2,351)	4,559 (1,993)	2,794 (639)	1,618	
F D 0 0 3	# of training profiles	10	30	70	All
	LSTM [11]	26,083 (34,446)	836 (394)	478 (269)	852
CNN+LSTM [29]	8,891 (13,737)	1,337 (330)	734 (360)	1,420	
CNN [10]	13,693 (2,430)	660 (110)	670 (71)	618	
Siamese (model)	10,189 (2,492)	698 (158)	494 (35)	304	
Siamese (reference)	9,175 (2,436)	628 (84)	396 (30)	270	
F D 0 0 4	# of training profiles	50	100	200	All
	LSTM [11]	34,999 (22,327)	39,715 (23,764)	40,690 (31,717)	5,550
CNN+LSTM [29]	62,820 (37,721)	20,580 (10,859)	29,904 (17,322)	4,630	
CNN [10]	34,127 (5,883)	16,083 (3,767)	8,415 (2,380)	5,861	
Siamese (model)	29,072 (5,458)	26,092 (2,184)	13,098 (2,013)	4,056	
Siamese (reference)	18,531 (7,308)	12,369 (3,380)	5,806 (768)	2,797	

F D 0 0 1	# of training profiles	10	30	70	All
	LSTM [11]	19.87 (2.79)	18.92 (7.14)	14.32 (1.32)	16.14
CNN+LSTM [29]	20.18 (2.96)	17.97 (0.91)	16.33 (1.03)	16.13	
CNN [10]	18.58 (1.78)	15.67 (0.89)	15.37 (0.68)	13.86	
Siamese (model)	17.55 (1.30)	15.56 (0.83)	14.81 (0.91)	12.34	
Siamese (reference)	16.71 (1.01)	14.36 (0.93)	13.49 (0.52)	12.47	
F D 0 0 2	# of training profiles	50	100	200	All
	LSTM [11]	24.29 (2.30)	24.84 (6.43)	24.45 (6.44)	24.49
CNN+LSTM [29]	26.83 (1.59)	26.25 (1.24)	25.26 (1.05)	20.46	
CNN [10]	24.52 (1.93)	22.01 (2.31)	20.29 (1.38)	18.91	
Siamese (model)	24.98 (1.41)	23.10 (1.18)	20.45 (1.16)	17.86	
Siamese (reference)	24.37 (1.21)	21.77 (0.99)	20.27 (0.90)	18.18	
F D 0 0 3	# of training profiles	10	30	70	All
	LSTM [11]	22.91 (4.26)	15.40 (1.11)	13.34 (1.09)	16.18
CNN+LSTM [29]	22.56 (3.28)	17.75 (0.86)	15.45 (0.98)	17.12	
CNN [10]	20.92 (4.19)	16.59 (0.88)	15.41 (0.88)	13.55	
Siamese (model)	20.08 (4.12)	15.88 (0.93)	13.84 (0.56)	12.77	
Siamese (reference)	19.40 (4.03)	14.99 (0.72)	13.19 (0.24)	11.88	
F D 0 0 4	# of training profiles	50	100	200	All
	LSTM [11]	26.99 (1.99)	26.34 (0.61)	26.21 (1.13)	28.17
CNN+LSTM [29]	30.13 (1.65)	27.14 (0.52)	27.66 (1.02)	23.26	
CNN [10]	28.85 (1.52)	26.49 (1.14)	23.90 (0.59)	23.32	
Siamese (model)	29.31 (1.18)	27.53 (1.18)	24.33 (0.76)	22.80	
Siamese (reference)	29.18 (1.27)	27.03 (1.07)	23.44 (0.86)	22.11	

Experimental results

- 최신 방법들과 성능 비교

- 동일한 데이터 환경을 맞추기 위해 모든 데이터 사용

- 대부분의 경우 제안 방법의 성능이 우수한 것으로 확인됨

- 그러나, attention 등 상위 모델과의 비교가 없고 객관적으로 높은 성능을 보이지는 않음

Dataset	FD001		FD002		FD003		FD004	
Metric	Score	RMSE	Score	RMSE	Score	RMSE	Score	RMSE
CNN [10]	360	13.86	3,352	18.91	618	13.55	5,861	23.32
LSTM [11]	338	16.14	4,450	24.49	852	16.18	5,550	28.17
MODBNE [39]	334	15.04	5,585	25.05	422	12.51	6,558	28.66
CNN+LSTM [29]	303	16.13	3,440	20.46	1,420	17.12	4,630	23.26
GA+RBM+LSTM [15]	231	12.56	3,366	22.73	251	12.1	2,840	22.66
BiLSTM-ED [40]	273	14.74	3,099	22.07	574	17.48	3,202	23.49
RULCLIPPER [41]	216	13.27	2,796	22.89	317	16.00	3,132	24.33
EN [6]	228	13.58	2,650	19.59	1,727	19.16	2,901	22.15
Proposed reference-based estimation	253	12.47	1,618	18.18	270	11.88	2,797	22.11

< Score / RMSE 성능 비교 >

Conclusion

- Siamese network 구조를 활용한 health representation learning 방법 제안
 - RUL의 변화와 health representation map의 변화 간의 일관성을 유지시켜 overfitting을 감소시킴
 - Reference-based 예측 방법을 더하여 추정치 분산을 줄이는 효과를 보임
- 제안된 학습 및 예측 방법이 RUL 예측 성능 뿐만 아니라 데이터 부족에 대한 robustness 향상에도 기여함을 실험 결과를 통해 보여줌
- Representation learning 기반의 수명 예측 논문들 중 해당 태스크에 맞게 잘 재구성한 방법론으로 보여짐
 - Feature 단에서의 loss 비교
 - RUL 특성이 잘 반영될 수 있는 representation map 구성
 - 그러나, attention 및 transformer 등 상위 모델보다는 현저히 낮은 성능으로 내부 네트워크 재구성을 통해 성능 향상 가능성이 있을 것으로 생각됨