Remaining useful life prediction on time series

2023년도 동계 세미나

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

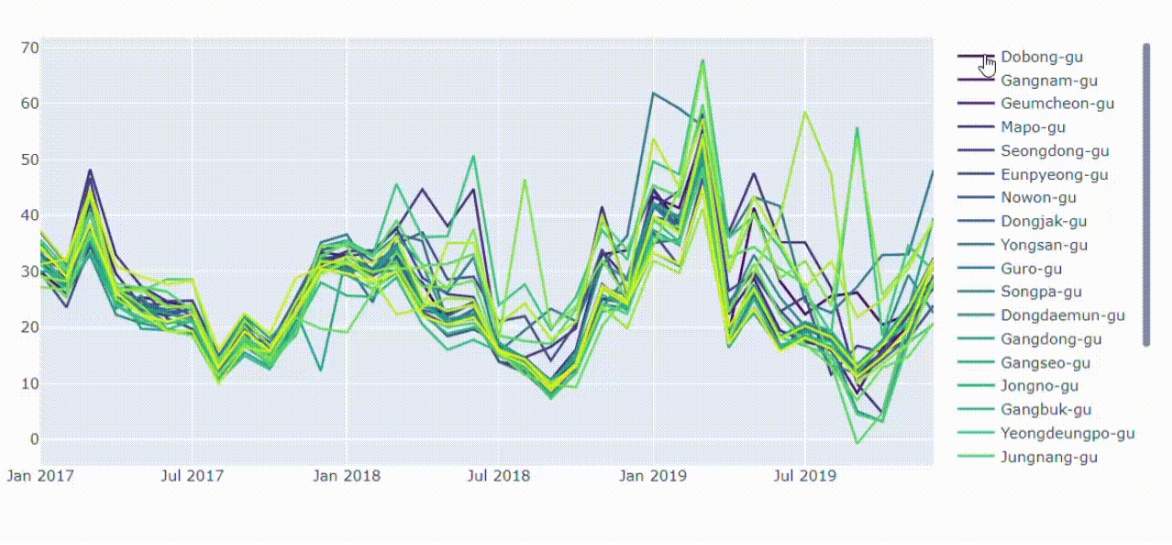
Presented By
Yein Park

Outline

- Background
  - Time series domain
  - Necessity of research on remaining useful life prediction
- Remaining useful life estimation via transformer encoder enhanced by a gated convolutional unit
- An integrated multi-head dual sparse self-attention network for remaining useful life prediction

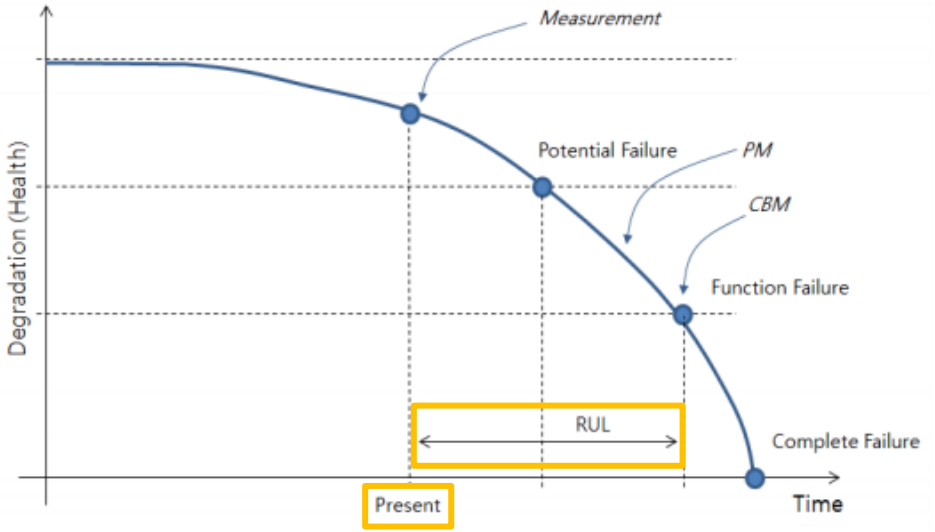
Background

- Time series domain
  - A series of sequential datasets collected over a period of time
    - Being ordered in terms of time
    - Consecutive observations correlated with each other
  - 시계열 데이터 분석의 목적
    - 과거 시점의 시계열 데이터가 갖고 있는 규칙성을 발견해 이를 모형화 / 모델링하는 것
    - 추정된 모형 / 모델을 통해 미래 시점을 예측하는 것



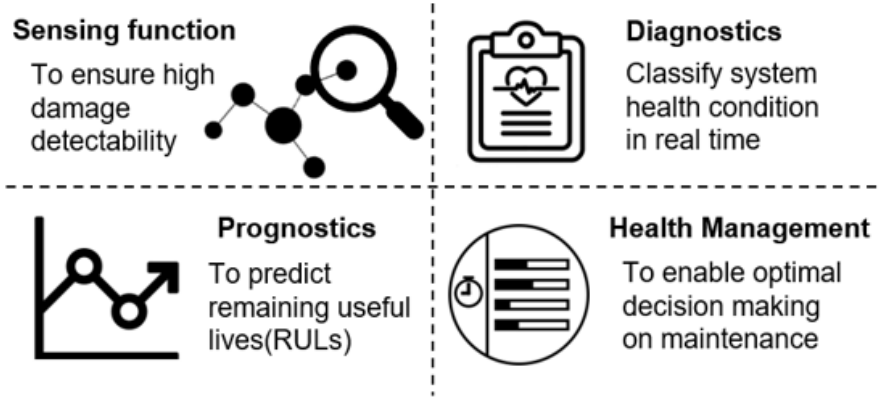
Background

- 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

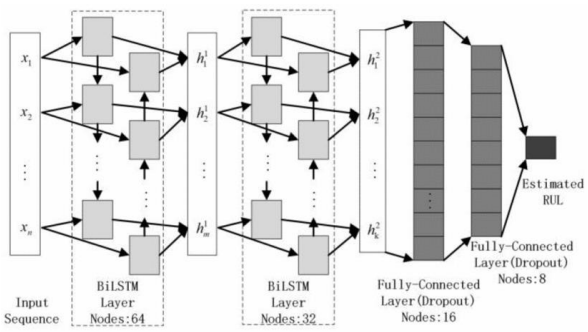
- Necessity of research on remaining useful life prediction
  - PHM (prognostic health management, 예후 상태 관리) 기술의 중요성 증가
    - 시스템 작동을 지속적으로 모니터링하고 장애 수준이나 사용할 수 없는 조건이 발생할 때 비정상을 진단
    - 유지관리 효율성, 안정 작동성, 시스템 성능 개선 면에서 중요성 증가
    - 해당 기술을 통해 필요한 경우에만 condition-based maintenance 수행할 수 있으며 유지 관리 비용을 크게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

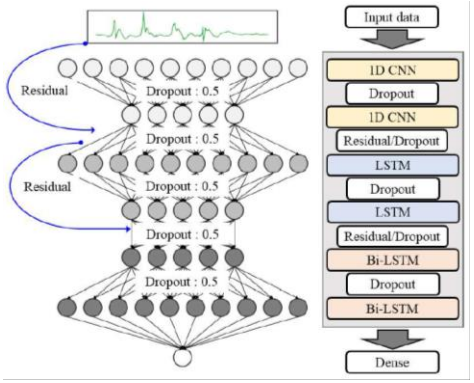


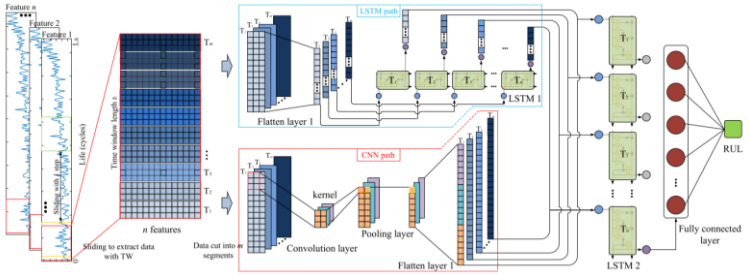
Prognostics and Health Management

Background

- Trends in remaining useful life prediction research
  - Single / sequential / parallel structure

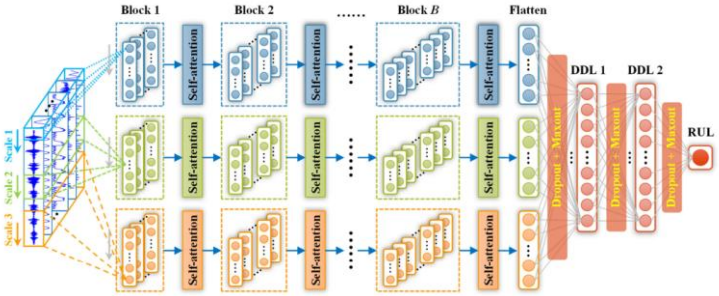


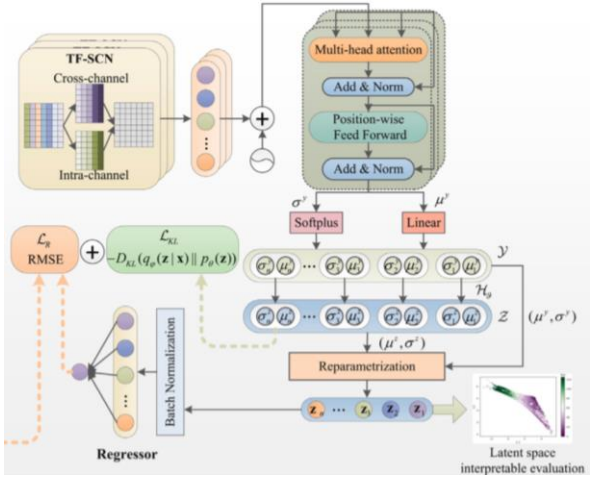




Background

- Trends in remaining useful life prediction research
  - Attention mechanism / **Transformer**

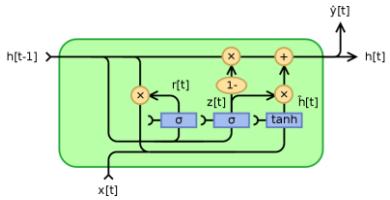




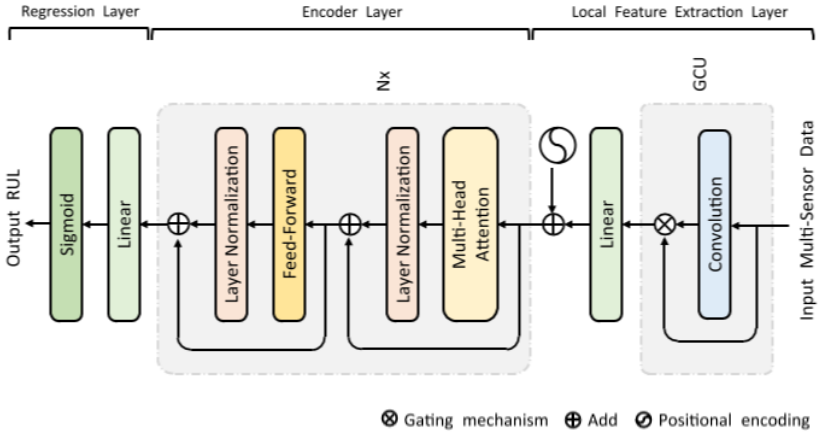
Mo, Yu, et al. "Remaining useful life estimation via transformer encoder enhanced by a gated convolutional unit." *Journal of Intelligent Manufacturing*, 2021.

Introduction

- 잔여수명 예측 분야에 있어 Transformer 구조를 적용한 최초의 논문
- Encoder layer: Transformer encoder를 모델 backbone으로 사용
  - Time sequence에서 long-term dependency 캡처
- Local feature extraction layer: gated convolutional unit (GCU) 제안
  - Gated recurrent unit (GRU)와 유사한 구조 및 의미로서, reset gate와 update gate로 구성됨
  - 입력 데이터에 대한 local feature 추출
  - Transformer encoder에서 local context에 둔감한 특성을 갖는 attention mechanism을 보완하기 위함



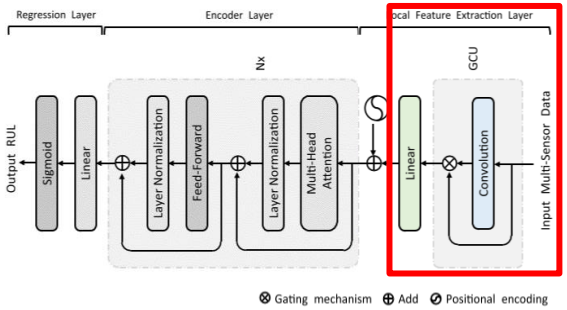
< **GRU** >



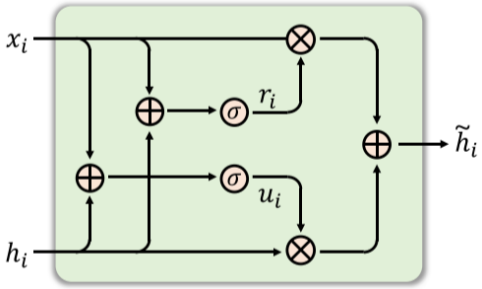
< **전체 프레임워크** >

Method

- 1) Local feature extraction layer
  - 입력 데이터에 대한 local feature 추출
  - 입력 데이터와 상위 layer feature 간 유익한 local feature 제공
  - **Gated convolutional unit (GCU)**
    - 입력 데이터에 CNN 적용
      - $h_i = conv([x_{i-k/2}, \dots, x_i, \dots, x_{i+k/2}])$ x_i: input data
    - 입력 데이터와 CNN feature간 gating mechanism 적용
      - Reset gate $\boldsymbol{r_i} = \sigma(W_r h_i + V_r x_i + b_r)$ σ:sigmoid function, W_r&V_r&b_r: parameters
        - → 입력 데이터의 정보를 얼마나 버릴지 조정하는 gate
      - Update gate $\boldsymbol{u_i} = \sigma(W_u h_i + V_u x_i + b_u)$ W_u&V_u&b_u: parameters
        - → 이전 레이어의 정보를 얼마나 유지할지 조정하는 gate
    - $\tilde{h}_i = h_i \otimes \boldsymbol{u_i} + x_i \otimes \boldsymbol{r_i}$ ⊗:element-wise multiplication
      - CNN ⊗ update 확률 + input ⊗ reset 확률
  - Linear mapping
    - GCU 출력 값인 $\tilde{h}_i$에 대하여 linear mapping 적용



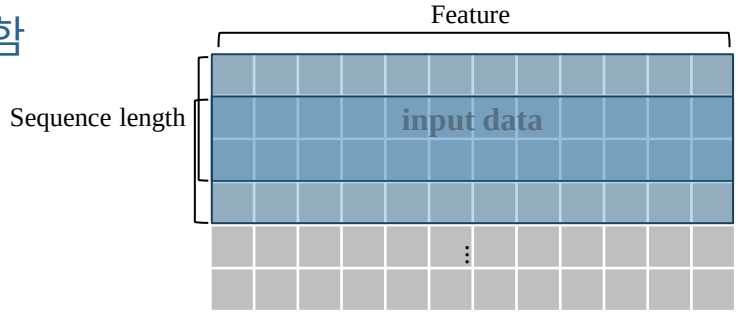
< Local feature extraction layer >



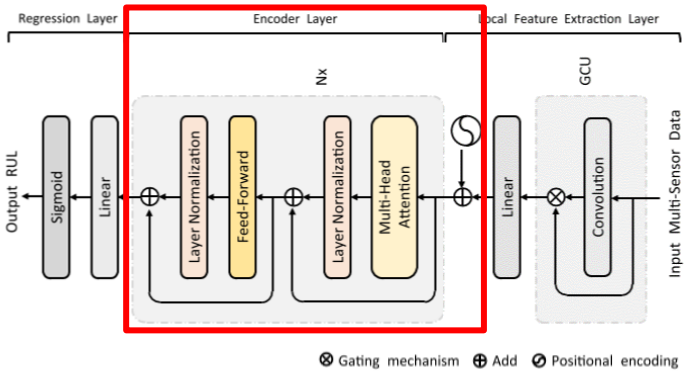
< GCU >

Method

- 2) Encoder layer
  - Local feature extraction layer의 출력 값을 입력으로 함
  - **Transformer encoder 사용**
    - Time sequence에서 long-term dependency 캡처
    - 커널 크기에 대한 제한이 없음
      - CNN 기반 방법은 커널에 대한 제한이 있음
    - 완전한 병렬 계산이 가능함
      - RNN 기반 방법은 과거 입력이 처리되어야 다음 입력으로 넘어가는 특성이 있음
      - 즉, 모든 입력의 처리 과정이 순차적으로 거쳐져야 하므로 시간 소모가 큰 문제가 있음



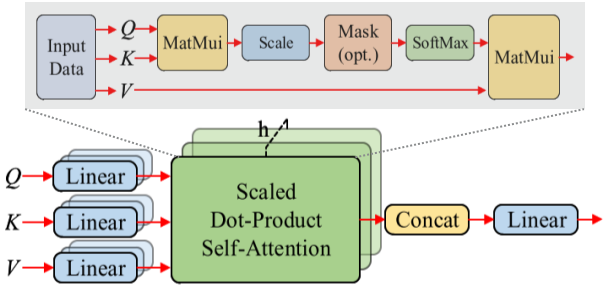
< 입력 데이터 shape >



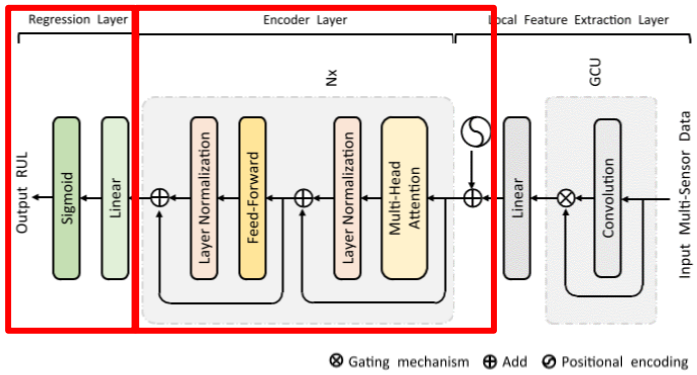
< Encoder layer >

Method

- 2) Encoder layer: Transformer encoder
  - 크게 position encoding, multi-head attention, fully-connected layer로 구성됨
  - Position encoding
    - 서로 다른 주파수의 사인, 코사인 함수 사용
    - 입력 데이터 각각의 고유한 토큰 위치 값 획득
      - Local feature extraction layer의 출력 값에 더해짐
  - Multi-head attention
    - Self-attention function이 적용된 여러 개의 헤드로 구성됨
  - Fully-connected layer
- 3) Regression layer
  - Fully-connected layer



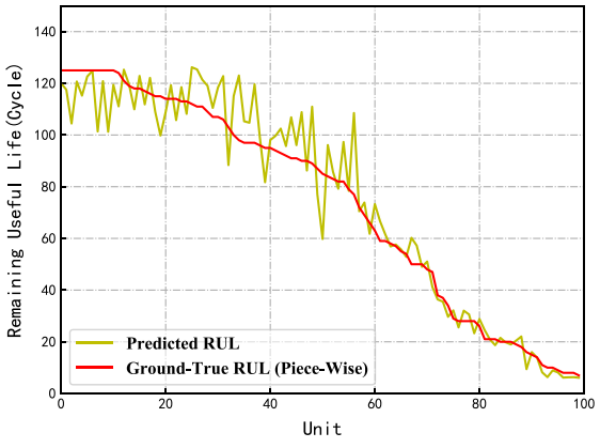
< Multi-head self-attention 과정 >

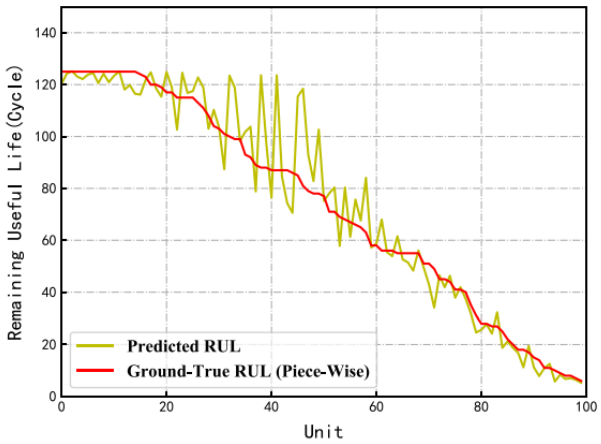


< Encoder layer, regression layer >

Results

- Comparison between the actual RULs and the estimated RULs



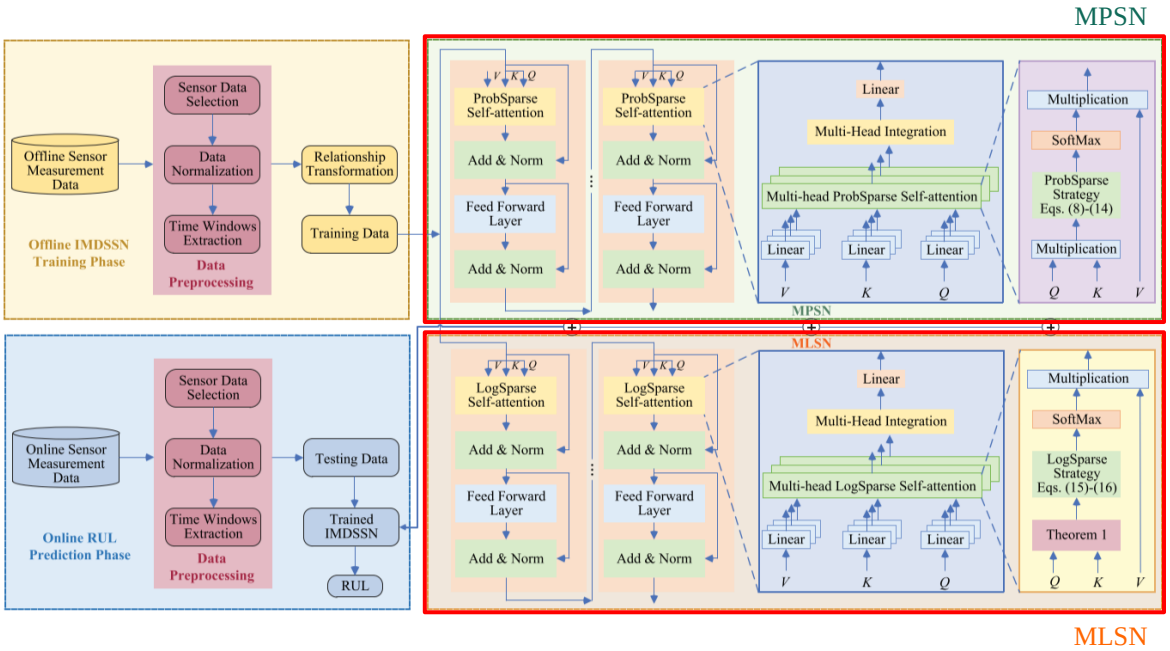


Method	FD001	FD002	FD003	FD004	Average
CNN (Badu *et al.*, 2016) Babu et al., (2016)	18.45	30.29	19.82	29.16	24.43
DBN (Zhang *et al.*, 2016) Zhang et al. (2016)	15.04	25.05	12.51	28.66	20.32
LSTM-FNN (Zheng *et al.*, 2017) Zheng et al. (2017)	16.14	24.49	16.18	28.17	21.25
CNN-FNN (Li *et al.*, 2018) Li et al. (2018)	12.61	22.36	12.64	22.43	**17.51**
Auto-Encoder (Yu *et al.*, 2019) Yu et al. (2019)	14.74	22.07	17.48	23.49	19.45
RBM-LSTM-FNN (Ellefsen *et al.*, 2019) Ellefsen et al. (2019)	12.56	22.73	12.10	22.66	**17.51**
DCNN-FNN (Xu *et al.*, 2020) Xu et al. (2020)	12.61	28.51	12.62	30.73	21.12
Auto-Encoder (Yu *et al.*, 2020) Yu et al. (2020)	13.58	**19.59**	19.16	**22.15**	18.62
GCU-Transformer (this paper)	**11.27**	22.81	**11.42**	24.86	17.59

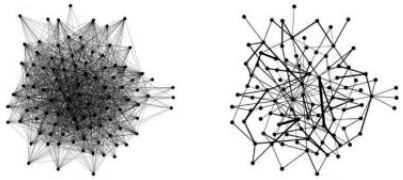
"An integrated multi-head dual sparse self-attention network for remaining useful life prediction." *Reliability Engineering & System Safety*, 2023.

Introduction

- 개선된 Transformer 기반의 multi-head dual sparse self-attention network 제안
  - Transformer의 한계인 계산 복잡성을 고려
  - Local key region에 집중하는 연구 필요
  - Multi-head ProbSparse self-attention network (MPSN)
    - Dot product operation에 있어 입력 데이터상의 주요 feature를 필터링하여 계산 효율성을 향상
  - Multi-head LogSparse self-attention network (MLSN)
    - Time window 길이에 따라 증가하는 계산 복잡성을 줄이기 위한 로그 기반 sparse strategy 제안



Method



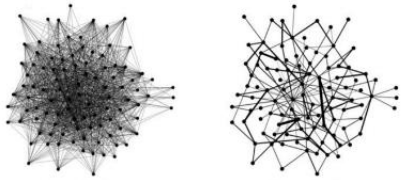
Dense ↔ Sparse

- 1) Multi-head ProbSparse self-attention network (MPSN)
  - Dot product operation의 주요 feature를 필터링하여 계산 효율성 향상
    - Self-attention mechanism의 dot product operation에 있어 probability distribution은 sparse할 것이라 가정
      - 즉, attention score의 probability distribution은 sparse 할 것임
      - 소수의 time point에서의 dot product operation만이 RUL 예측에 상당한 영향을 미침
      - 중요한 역할을 하는 dot product operation을 필터링하여 연산 효율을 향상시킴
    - MPSN 연산 과정

MPSN의 dot product operation	기존 dot product operation
$A_i(q_i, K, V) = \sum_j \frac{k(q_i, k_j)}{\sum_l k(q_i, k_l)} v_j = E_{p(k_j \mid q_i)}[v_j]$	$Attention(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$

      - 기존 dot product operation은 computational load가 큼
      - MPSN - conditional probability $p(k_j|q_i)$: query q_i가 주어졌을 때 key k_j일 확률
        - ✓이를 통해, dot product operation에 있어 key의 일부와 query만을 취함
      - $k(q_i, k_j)$: $\exp\left(q_i k_j^T / \sqrt{d_k}\right)$ 으로 exponential function임

Method

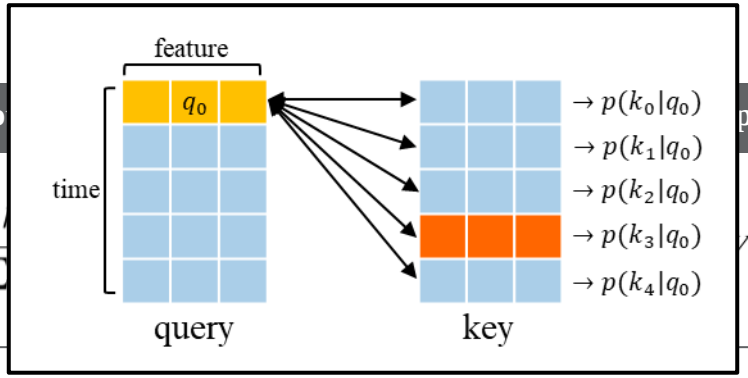


Dense ↔ Sparse

- 1) Multi-head ProbSparse self-attention network (MPSN)
  - Dot product operation의 주요 feature를 필터링하여 계산 효율성 향상
    - Self-attention mechanism의 dot product operation에 있어 probability distribution은 sparse할 것이라 가정
      - 즉, attention score의 probability distribution은 sparse 할 것임
      - 소수의 time point에서의 dot product operation만이 RUL 예측에 상당한 영향을 미침
      - 중요한 역할을 하는 dot product operation을 필터링하여 연산 효율을 향상시킴
    - MPSN 연산 과정

MPSN의 do

$$A_i(q_i, K, V) = \sum_j \frac{}{\sum}$$



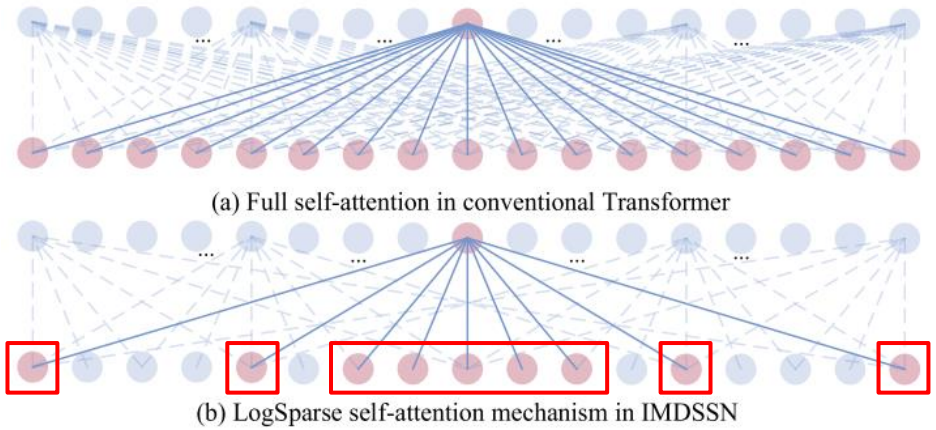
product operation

$$V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

      - 기존 dot product operation은 computational load가 큼
      - MPSN - conditional probability $p(k_j|q_i)$: query q_i가 주어졌을 때 key k_j일 확률
        - ✓이를 통해, dot product operation에 있어 key의 일부와 query만을 취함
      - $k(q_i, k_j)$: $\exp\left(q_i k_j^T / \sqrt{d_k}\right)$ 으로 exponential function임

Method

- 2) Multi-head LogSparse self-attention network (MLSN)
  - Time window 길이에 따라 증가하는 복잡성을 줄이기 위한 로그 기반 sparse strategy 제안
    - MPSN과 유사하게, MLSN도 multi-head self-attention mechanism network 기반임
    - 기존의 Transformer model은 주어진 time window length (L)에 대하여 full self-attention이 수행됨
      - Cell index set $I_l^k = \{1,2,\dots,l,\dots,L\}$
      - 입력 데이터 차원 L이 증가함에 따라 계산 복잡성이 제곱꼴로 증가함
    - 계산 복잡성을 완화하기 위해 cell index set 재구성
      - $I_l^k = \{l-2^{[\log_2 l]}, l-2^{[\log_2 l]-1}, \dots, l-2^0, l, l+2^0, \dots, l+2^{[\log_2 l]-1}, l+2^{[\log_2 l]}\}$
        - ✓입력 데이터 차원 L이 증가함에 따라 계산 복잡성이 두배로 증가함

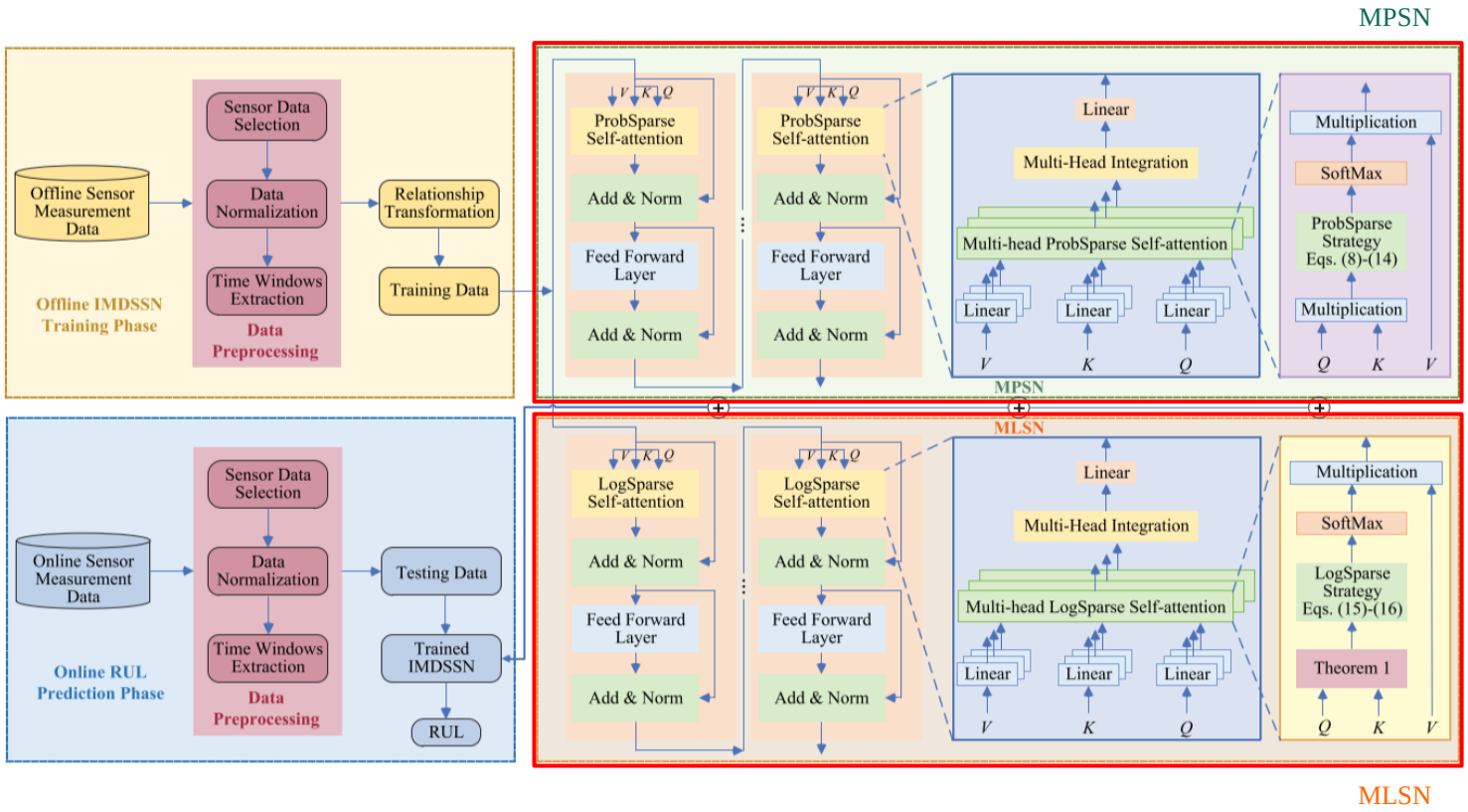


(a) Full self-attention in conventional Transformer

(b) LogSparse self-attention mechanism in IMDSSN

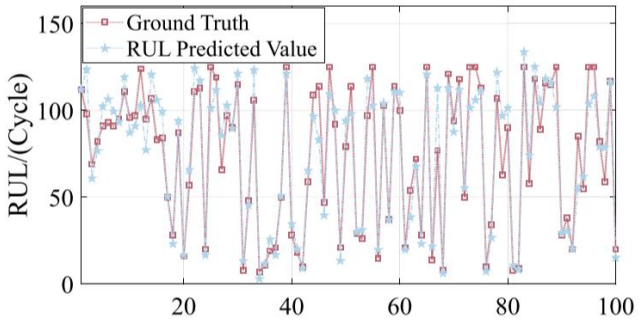
Method

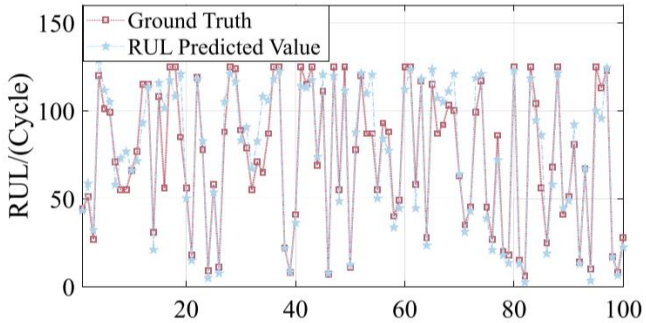
- 개선된 Transformer 기반의 multi-head dual sparse self-attention network 제안
  - MPSN와 MLSN의 output feature가 concatenation되어 최종 예측된 RUL 값 취득



Results

- Comparison between the actual RULs and the estimated RULs





Approach	RMSE				S-Score			
	FD001	FD002	FD003	FD004	FD001	FD002	FD003	FD004
SVM [32]	40.72	52.99	46.32	59.96	7703.33	316483.31	22541.58	141122.19
CNN [12]	18.45	30.29	19.82	29.16	1286.70	13570.00	1596.20	7886.40
ELM [32]	17.27	37.28	18.90	38.43	523.00	498149.97	573.78	121414.47
MLP [32]	16.78	28.78	18.47	30.96	560.59	14026.72	479.85	10444.35
CNN-LSTM [11]	16.13	20.46	17.12	23.26	303.00	3440.00	1420.94	4630.00
DBN [32]	15.21	27.12	14.71	29.88	417.59	9031.64	442.43	7954.51
MODBNE [32]	15.04	25.05	12.51	28.66	334.23	5585.34	421.91	6557.62
CEED+DLSTM [33]	14.72	29.00	17.72	33.43	262.00	6953.00	452.00	15069.00
BiLSTM-ED [34]	14.74	22.07	17.48	23.49	273.00	3099.00	574.00	3202.00
IESGP [35]	14.72	24.81	14.99	28.61	331.90	4245.40	355.20	6280.80
Attention-LSTM [36]	14.53	–	–	27.06	322.44	–	–	5649.14
CNN-LSTM-DA [37]	14.40	27.23	14.32	26.69	290.00	9869.00	316.00	6594.00
BiGRU [13]	–	26.54	–	29.13	–	6352.00	–	6339.00
BiLSTM [38]	13.65	23.18	13.74	24.86	295.00	4130.00	317.00	5430.00
LSTM [18]	13.52	24.42	13.54	24.21	431.70	14459.00	347.30	14322.00
Transformer	13.32	19.83	13.92	21.88	329.00	3394.25	431.34	3566.14
AdaBN-DCNN [39]	13.17	20.87	14.97	24.57	279.00	2020.00	817.00	3690.00
MPSN	13.16	18.06	14.09	21.36	270.84	2168.99	410.73	3216.68
BiGRU-TSAM [31]	12.56	18.94	12.45	20.47	213.35	2264.13	232.86	3610.34
MLSN	12.50	18.30	13.57	22.65	245.76	2561.82	316.89	4056.62
Proposed IMDSSN	**12.14**	**17.40**	**12.35**	**19.78**	**206.11**	**1775.15**	**229.54**	**2852.81**

Conclusion

- 시계열 데이터 기반 잔여수명 예측 태스크
  - 다양한 Transformer 기반 구조 채택 증가
    - 그러나, 성능 측면에서 우수하지 못하다는 단점이 있음
    - 현재까지는 self-attention 기반 구조의 방법들의 성능이 더 우수함을 확인
  - 단순한 적용을 뛰어넘어 해당 태스크에 최적화된 구조로 변형 시도
    - 공장 및 산업체에서 실제 많은 적용이 되고 있는 태스크
    - 실시간 환경을 고려하여 계산 복잡성을 줄이기 위한 방법이 연구되고 있음