

중대사고 분야 인공지능의 적용성 및 활용가능성 연구 현황

# 중대사고 데이터의 시계열분석을 위한 딥러닝 적용방안

이대영

2024. 05. 08



|주|미래와도전  
FNC Technology Co., Ltd.

# CONTENTS

- 1 과제 개요
- 2 배경 지식
- 3 과제 수행 내용
- 4 Considerations

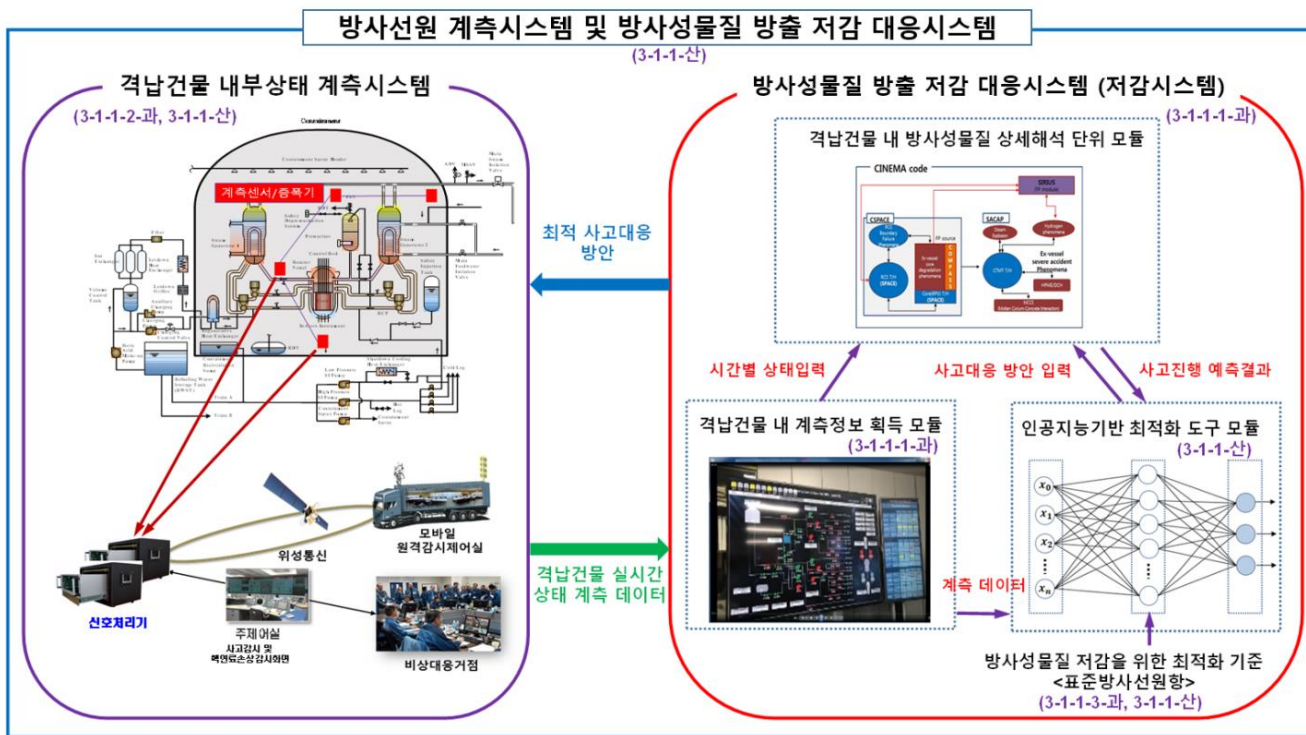
# 1

## 과제 개요

# 1 과제 개요

## (3-1-1) 중대사고 시 격납건물 내 방사선원 계측기술 개발

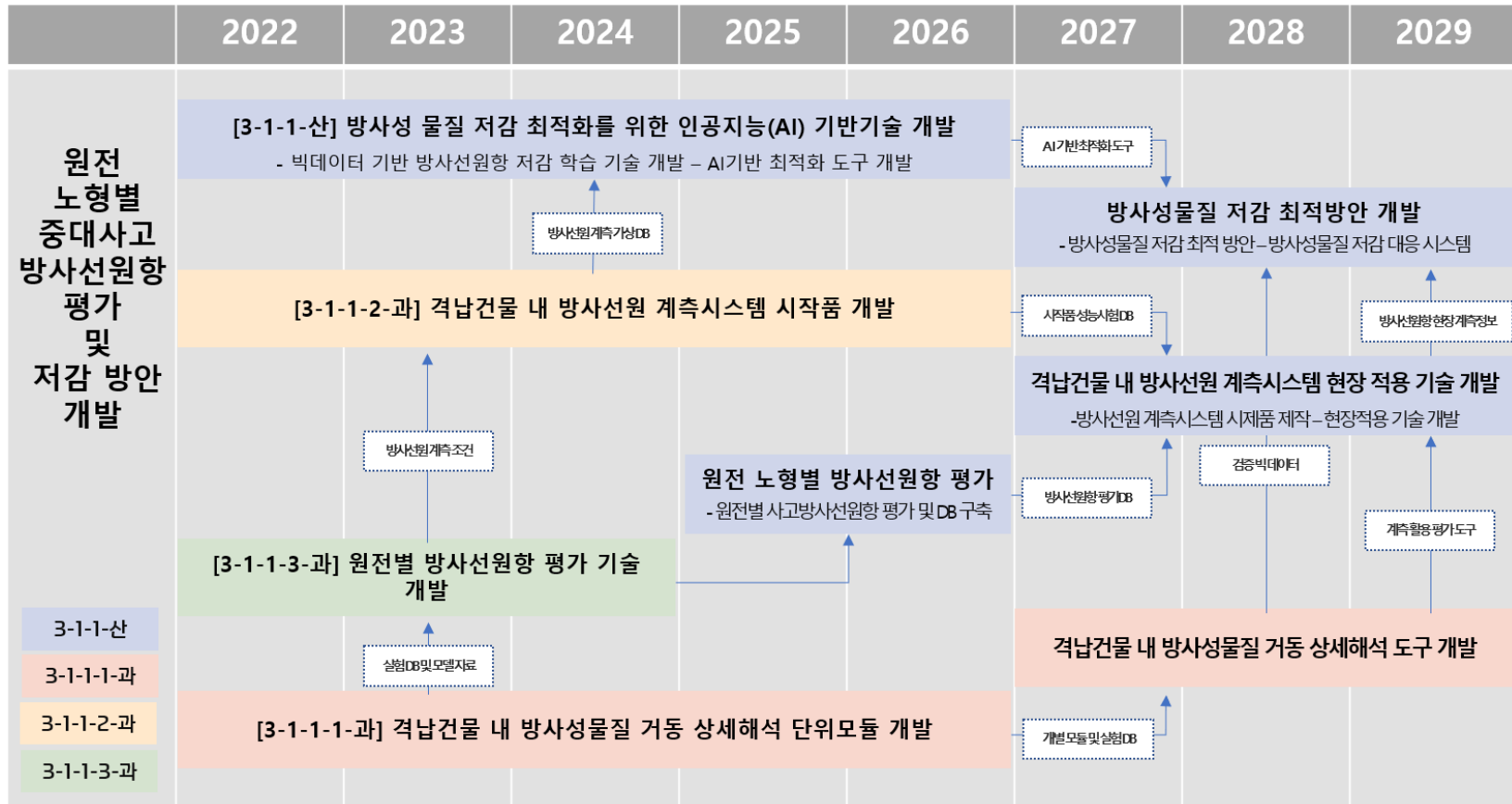
- (3-1-1-1 과기부) 중대사고시 격납건물 내 방사성물질 거동 상세해석 도구 개발
- (3-1-1-2 과기부) 중대사고시 격납건물 내 방사선원 계측시스템 시작품 개발
- (3-1-1-3 과기부) 중대사고시 격납건물 내 표준 방사선원항 평가기술 개발
- (3-1-1 산업부) 원전 노형별 중대사고 방사선원 평가 및 저감 방안 개발





# 1 과제 개요

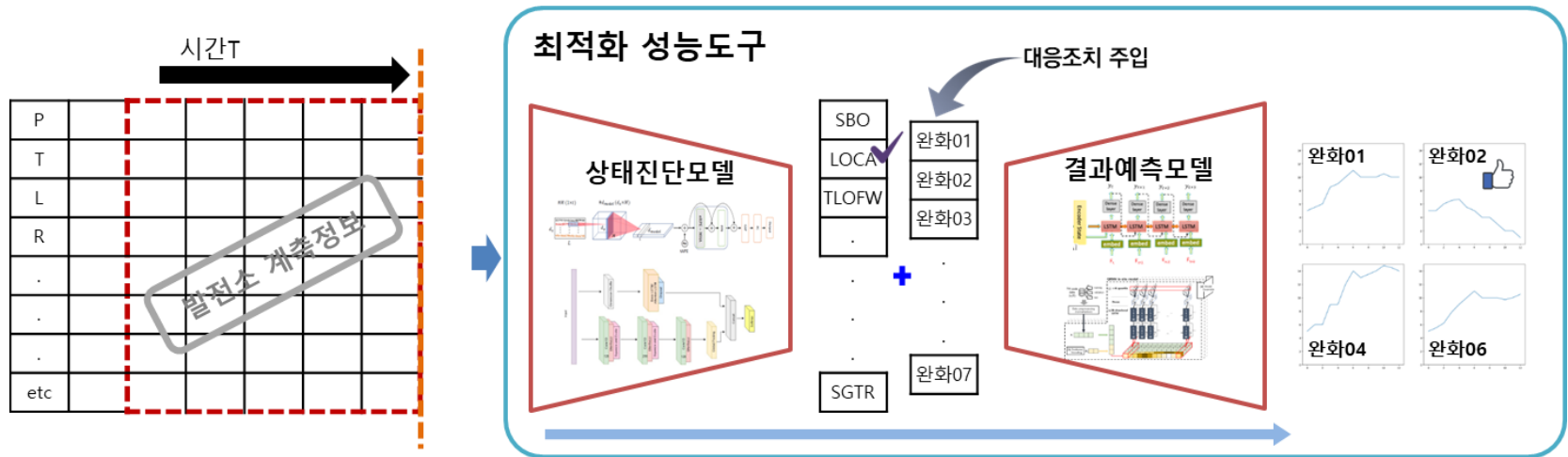
## 과제 연계사항 및 단계별 추진 일정



# 1 과제 개요

## 본 과제에서 인공지능 모델의 역할

- ▶ 상태진단 모델 : 발전소의 사고상태를 진단하고 이에 따른 대응 조치를 제시하는 모델
- ▶ 결과예측 모델 : 해당 대응조치를 통해 앞으로 발전소 내 방사성 물질이 저감됨을 검증하는 모델



## ▶ 필요기술 : 시계열 데이터 분석 기술

- 데이터를 통해 어떻게 발전소 상태를 진단할 수 있을까? → Time-Series Classification(TSC)
  - 초기사건 후 중대사고 진입 전까지 상황에 따라 운전원 조치가 다름
- 운전원 조치에 따라 과연 앞으로 Csl 농도는 낮아질까? → Time-Series Forecasting(TSF)
- 시계열 데이터를 분석하는 방법들은 기존 통계적 기법에서 딥러닝 방법론으로 전환 중

# 2

## 배경 지식

# 2 Background Knowledge

## 시계열 데이터의 이해

- ▶ 시간의 흐름에 따라 관측값이 변하는 데이터 : 미래를 알고 싶은 욕구
- ▶ 전통 통계학으로 데이터 분석을 하던 시절, 시간이라는 속성에 의해 오류 발생
- ▶ 데이터의 평균/분산과 같은 속성이 시간의 흐름에 따라 변함
- ▶ Non-Stationary(비정상성)
- ▶ 통계학적 시계열 데이터 분석 방법

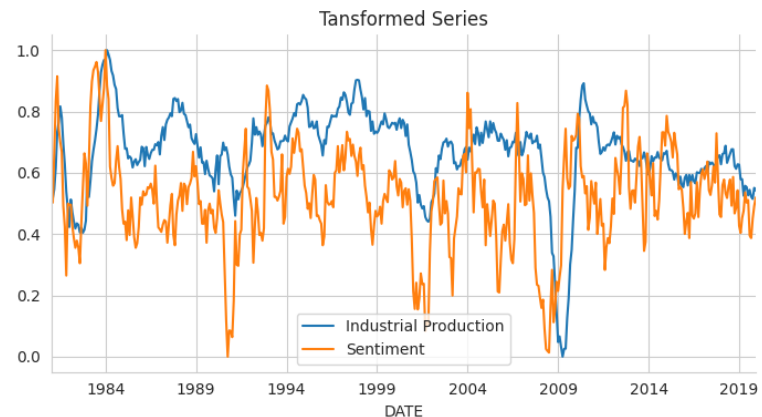
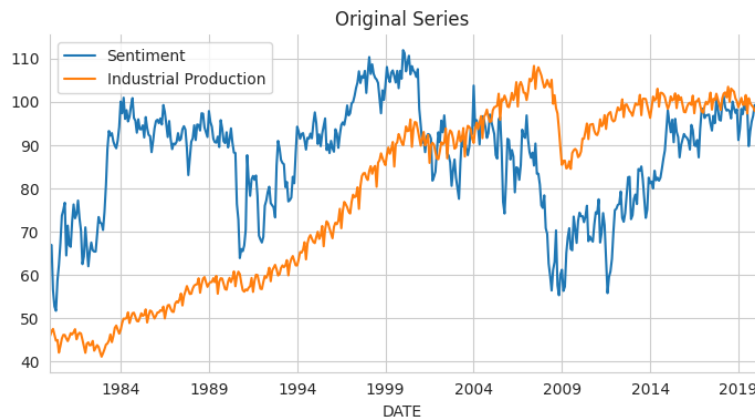
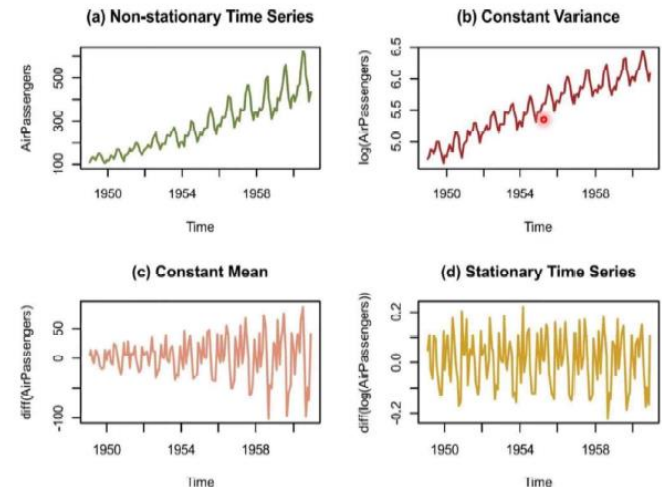
■ Feature Transform : Non-Stationary → Stationary

■ Differencing, Log Transform

■ Feature Adding

■ 선후관계를 갖는 시계열 추가(granger causality)

■ 공적분(cointegration) 관계의 시계열 추가





# 2 Background Knowledge

## 시계열 데이터의 이해

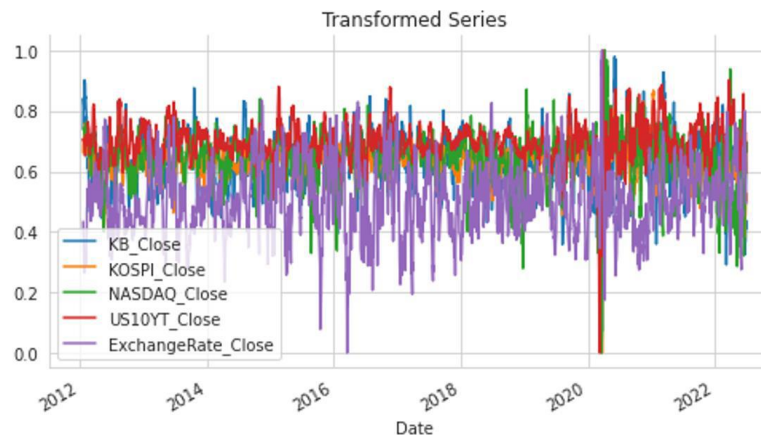
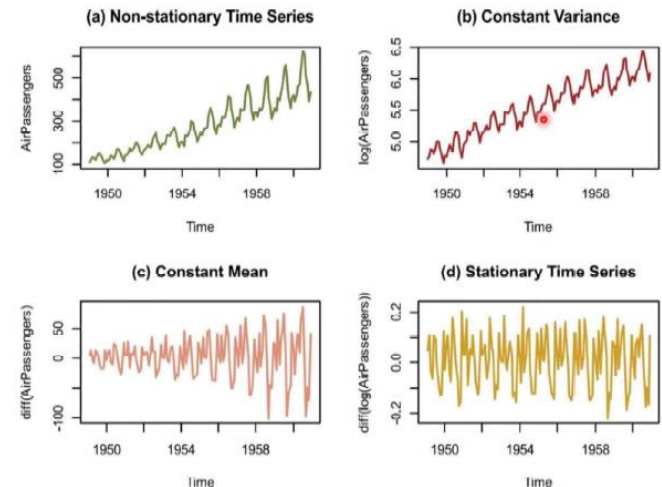
- 시간의 흐름에 따라 관측값이 변하는 데이터 : 미래를 알고 싶은 욕구
- 통계적 관점으로 데이터 분석을 하던 시절, 시간이라는 속성에 의해 오류 발생
- 데이터의 평균/분산과 같은 속성이 시간의 흐름에 따라 변함
- Non-Stationary(비정상성)
- 통계학적 시계열 데이터 분석 방법

■ Feature Transform : Non-Stationary → Stationary

■ Differencing, Log Transform

■ Feature Adding

- 선후관계를 갖는 시계열 추가(granger causality)
- 공적분(cointegration) 관계의 시계열 추가



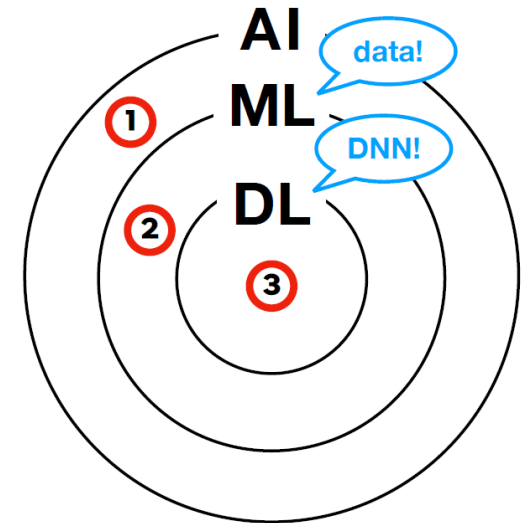
Feature를 추가할수록  
관계성을 찾기가 어려워짐

## 2 Background Knowledge

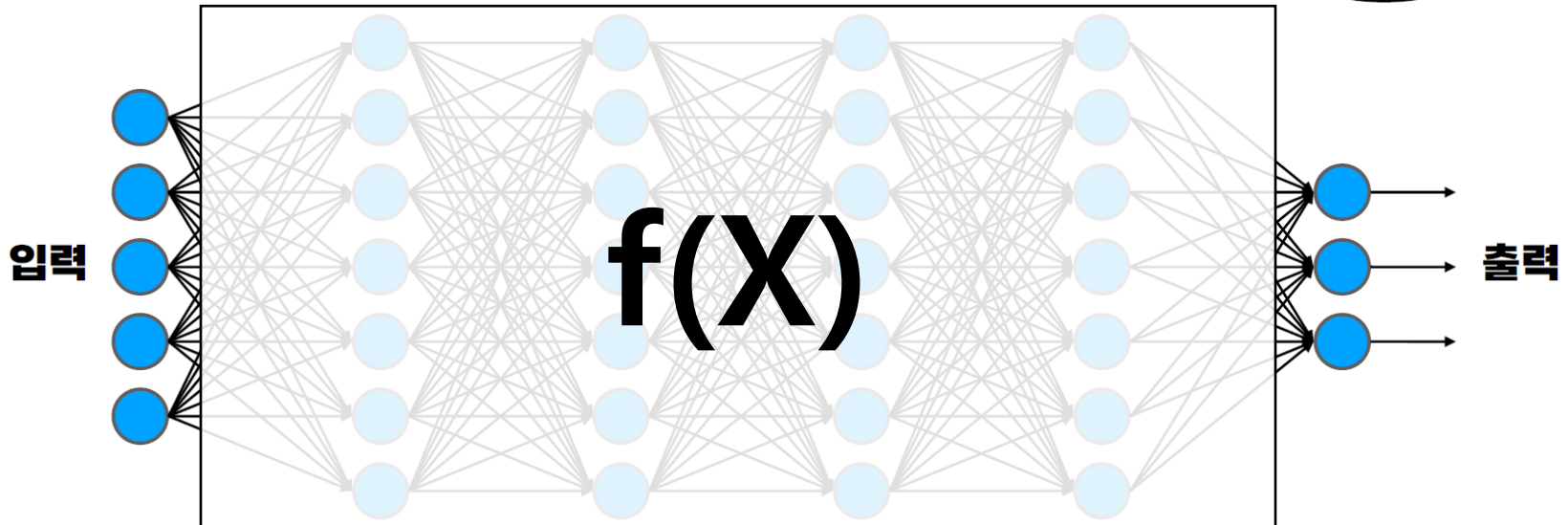
### 인공지능 기술의 분류

#### ➤ AI vs ML vs DL

- ① AI : 규칙 기반 알고리즘
- ② ML : 결정 트리, 선형회귀, XGBoost, SVM (Support Vector Machine)
- ③ DL : CNN, RNN, GAN, 트랜스포머, ...



#### ➤ Deep Learning

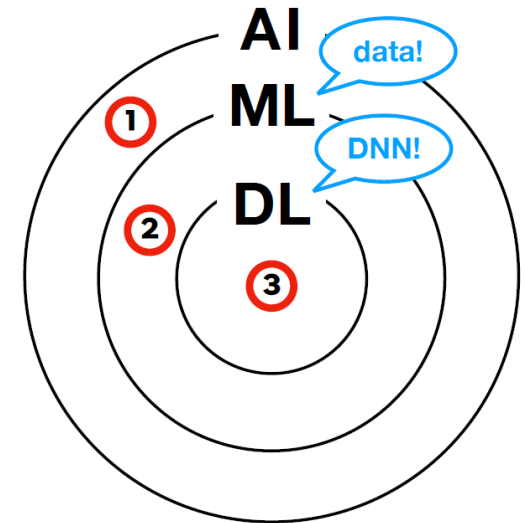


## 2 Background Knowledge

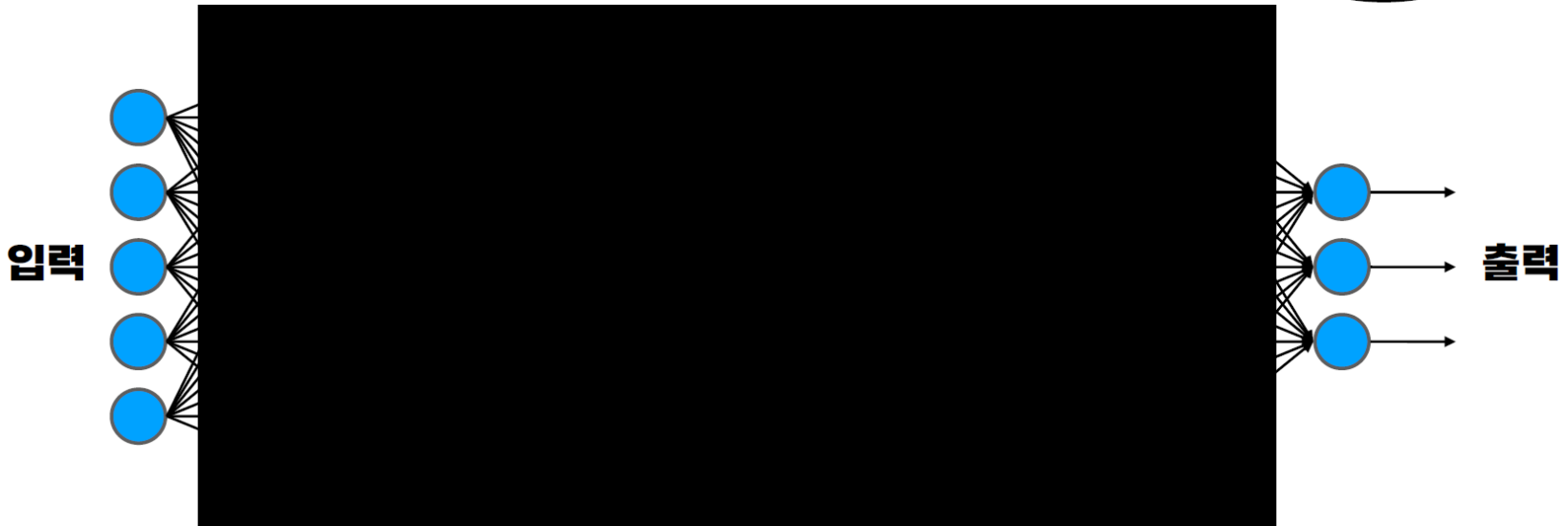
### 인공지능 기술의 분류

#### ➤ AI vs ML vs DL

- ① AI : 규칙 기반 알고리즘
- ② ML : 결정 트리, 선형회귀, XGBoost, SVM (Support Vector Machine)
- ③ DL : CNN, RNN, GAN, 트랜스포머, ...



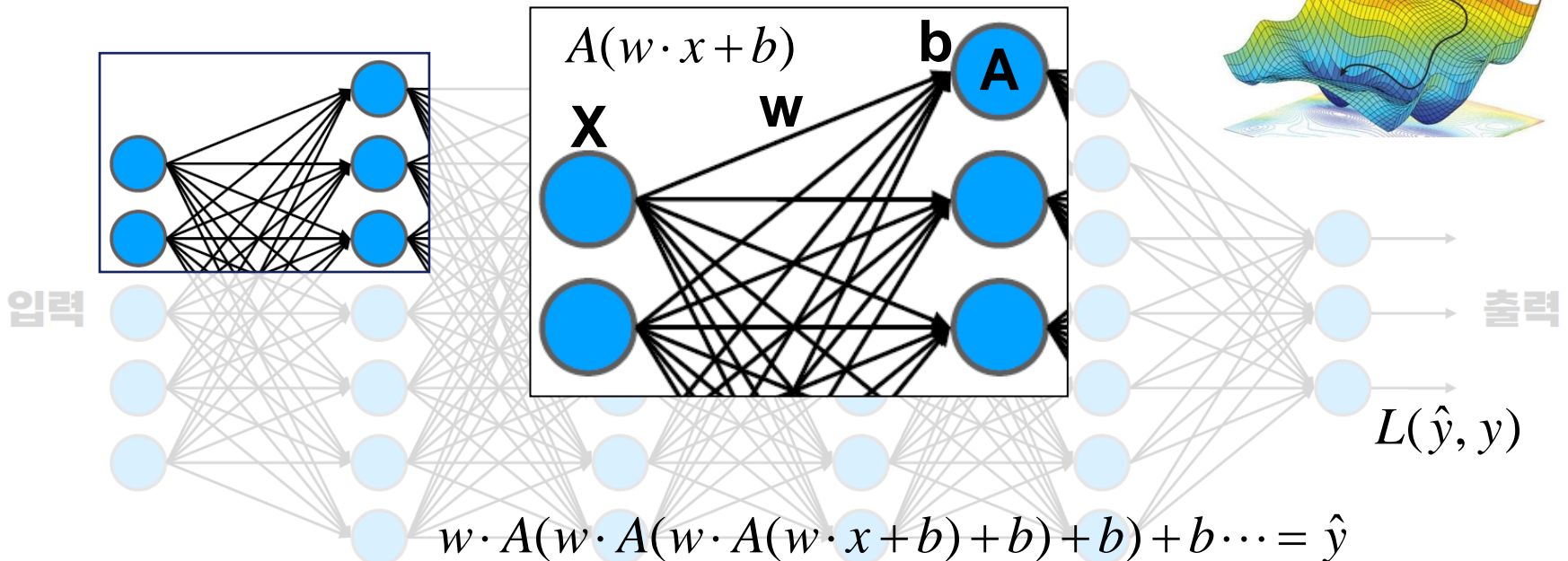
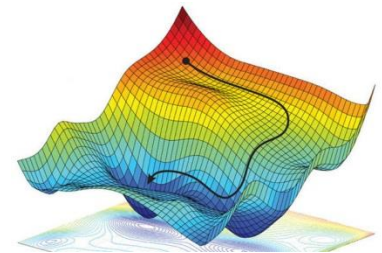
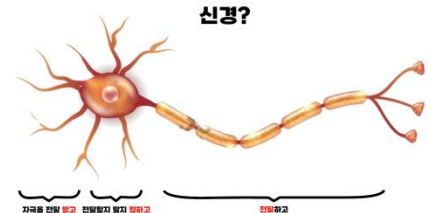
#### ➤ Deep Learning



# 2 Background Knowledge

## Artificial Neural Network의 원리

- 노드(혹은 unit)와 엣지(혹은 connection)로 이루어져있다.
- 웨이트(중요도,  $w$ )를 곱하고 바이어스(민감도,  $b$ )와 함께 더하고, 액티베이션!
- 여러가지 액티베이션이 존재. Linear, Sigmoid, ReLU, tanh ...
- 주어진 입력에 대해 원하는 출력이 나오도록 웨이트, 바이어스를 정해줘야 한다.
- 여기서, AI가 적절한 웨이트( $w$ ), 바이어스( $b$ )를 알아냄! (이것을 “AI가 학습한다” 라고 표현)
- 곱하고 더하고 activation, 곱하고 더하고 activation, .. 반복



# Background Knowledge

## Deep-Learning 모델

- 학습이란? 결국, 각 노드에 걸려있는 입력과 출력을 통해 최적의  $w$ 와  $b$ 를 찾는 과정
- 모델이란? 학습을 통해 결정된 파라미터(Weight, Bias)로 구성된 Matrix의 조합

```
Parameter containing:
tensor([[ 0.4223,  1.0829, -0.7585, ...,  1.3011, -0.2246, -0.6516],
        [ 3.3340, -0.5697, -1.1163, ..., -13.4995, -3.8654, -1.6603],
        [ 0.1018,  0.1551, -3.5475, ...,  5.5360, -0.5984,  3.6319],
        ...,
        [ 0.0933,  0.9214, -0.1964, ...,  4.7361, -1.0311,  0.7439],
        [ 1.5408,  0.3737, -1.5125, ...,  4.6930, -0.6784,  1.1959],
        [ 0.6573,  0.2671, -2.0278, ...,  1.8184,  2.6442, -0.4709]],
        device='cuda:0', requires_grad=True)
Parameter containing:
tensor([[ 0.0003, -0.0174,  0.0091, ..., -0.0149, -0.0151,  0.0066],
        [ 0.0276,  0.0565, -0.0226, ...,  0.0458,  0.0060,  0.0606],
        [-0.0084,  0.0156,  0.0493, ...,  0.0095,  0.0267,  0.0108],
        ...,
        [ 0.0064, -0.0641, -0.0178, ...,  0.0659,  0.0482, -0.0747],
        [-0.0164, -0.0333,  0.0448, ...,  0.0113,  0.0405,  0.0243],
        [-0.0396,  0.0035, -0.0345, ..., -0.0434,  0.0033, -0.0482]],
        device='cuda:0', requires_grad=True)
```



[Geoffrey Hinton]

- 이러한 기본적인 구조(MLP)에서 다양한 연산기법(정규화, 내적, Convolution 등)을 도입하여 고도화
- 전통적 방식(수치해석, 통계적)의 비선형 문제 풀이 방법을 데이터와 딥러닝 기술로 개선
- Computer Vision(CV), Natural Language Processing(NLP) : 데이터가 많다.
- GPT3 모델의 파라미터 수 : 1750억개(175B) / 45TB text data

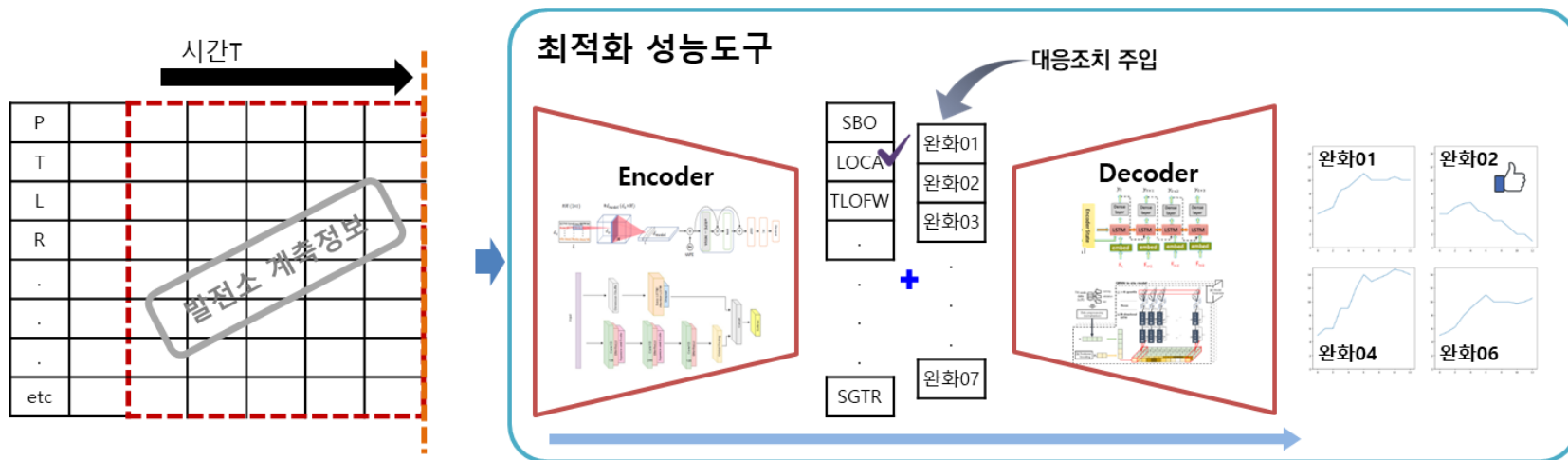


# 3

## 과제 수행 내용

## 본 과제에서 인공지능 모델의 역할

- ▶ 상태진단 모델 : 데이터를 통해 상황을 인지하여 발전소의 사고상태를 분류(Encoder)
- ▶ 결과예측 모델 : 대응조치에 따른 격납건물 방사성물질 장기 예측(Decoder)



## ▶ Encoder-Decoder 구조 : 인간의 생성과정을 재현

- 자연어 처리 번역기에 주로 사용
- Sequence-to-Sequence는 입력된 시퀀스로부터 다른 도메인의 시퀀스를 출력
- Encoder : 문맥을 이해하는 역할 → 벡터를 통해 상황을 인지(Understanding)
- Decoder : 이해한 내용을 바탕으로 Output을 생성 → 인지를 바탕으로 관련된 벡터를 생성(Generating)

## 상태진단 모델 개발

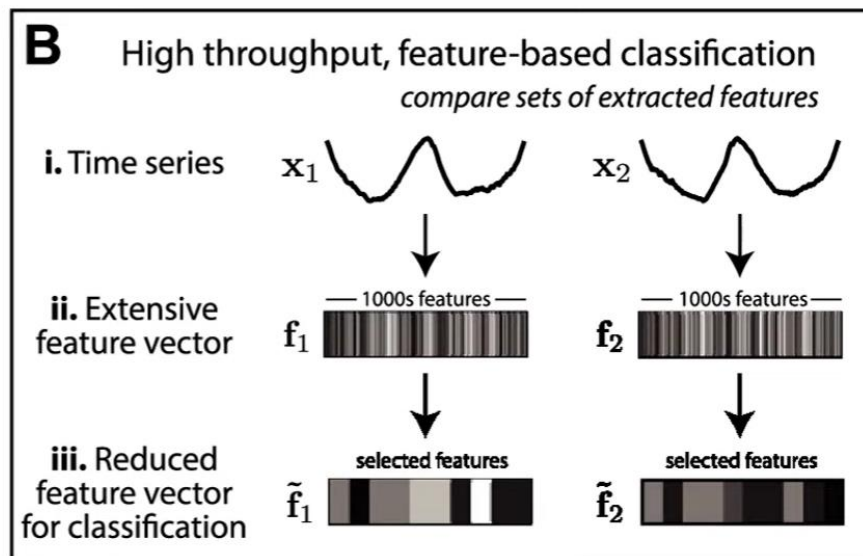
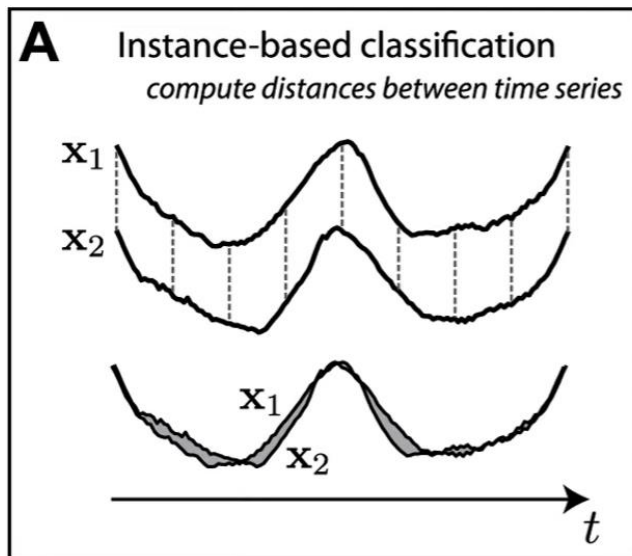
### Time-Series Classification 대표 방법론

#### Distance-based method(거리 기반 방법론)

- 특정 metric을 기반으로 시계열 데이터 간 거리를 계산하여 가까운 거리의 Class로 분류

#### Feature-based method(피처 기반 방법론)

- 시계열 데이터의 내재적인 특징을 추출하여 분류에 활용
- 시계열 데이터 자체의 내재적인 특징을 포착하는 것은 Challenging 함 -> Encoder

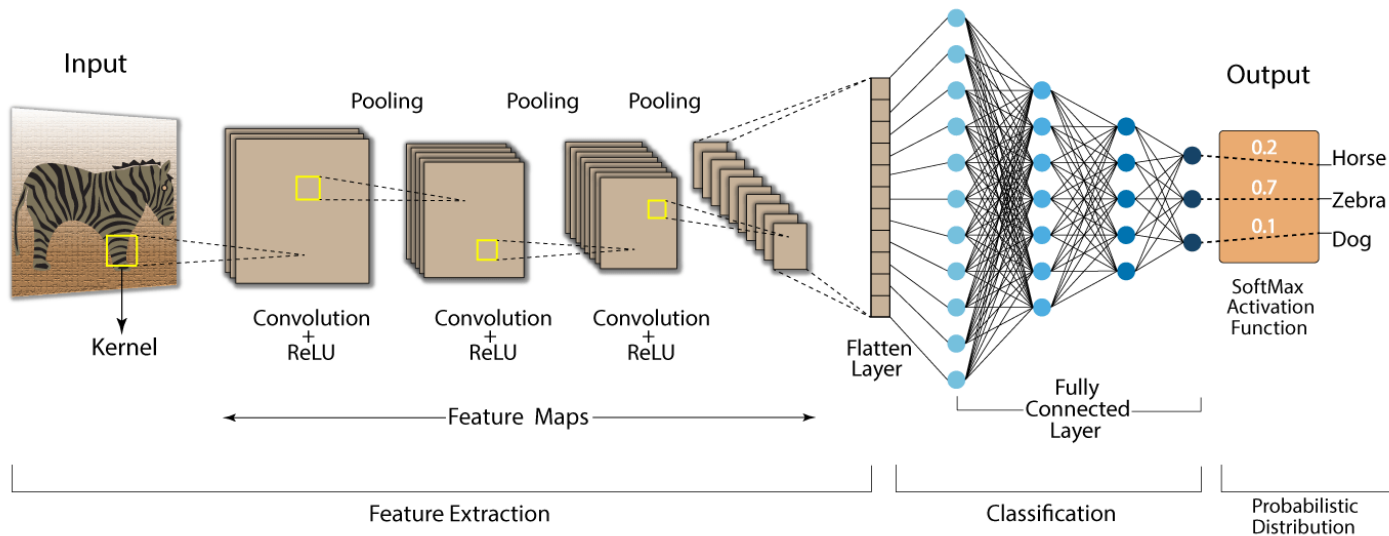
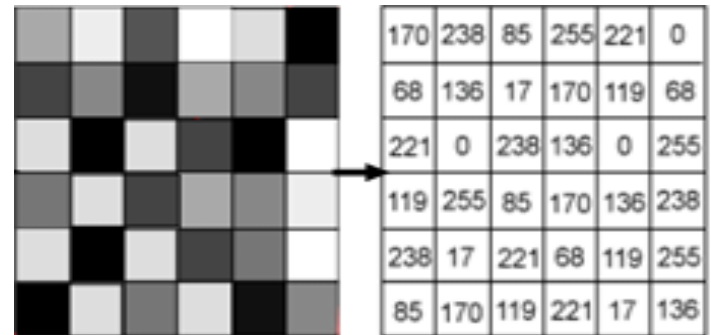
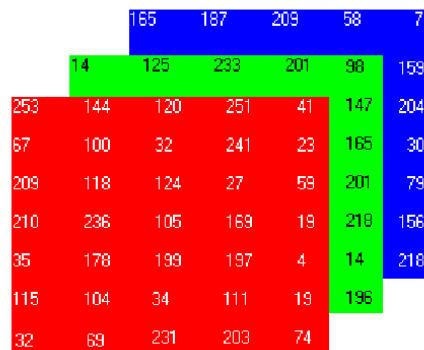
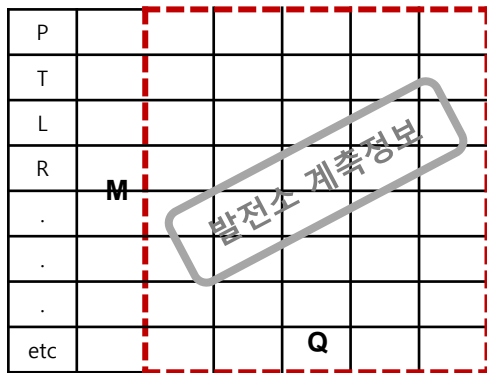


# 과제 수행 내용

## 상태진단 모델 개발

### Base Tech

#### CNN(Convolutional Neural Network)



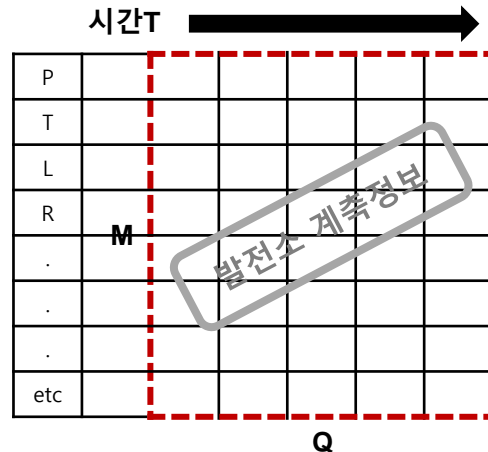
Sequential한  
상관관계를  
반영하지 못함.

## 상태진단 모델 개발

### Base Tech

#### 풀어야 하는 문제 : Seq2seq 문제

- 자연어 처리 분야에서 빠르게 발전함.
- 자연어는 Tokenizer와 Word Embedding을 통해 벡터화 됨.
- 자연어도 시간 순서에 따라 사용되므로 S2S 문제 라고 할 수 있음.
- 발전소 계측정보 [MxQ] 매트릭스와 자연어 벡터[MxQ] 매트릭스는 시간에 따른 벡터의 변화라는 측면에서 동일한 문제라고 할 수 있음.



Token String	Token ID	Embedded Token Vector
'<s>' ->	0 ->	[ 0.1150, -0.1438, 0.0555, ... ]
'<pad>' ->	1 ->	[ 0.1149, -0.1438, 0.0547, ... ]
'</s>' ->	2 ->	[ 0.0010, -0.0922, 0.1025, ... ]
'<unk>' ->	3 ->	[ 0.1149, -0.1439, 0.0548, ... ]
'.' ->	4 ->	[ -0.0651, -0.0622, -0.0002, ... ]
' the' ->	5 ->	[ -0.0340, 0.0068, -0.0844, ... ]
',' ->	6 ->	[ 0.0483, -0.0214, -0.0927, ... ]
' to' ->	7 ->	[ -0.0439, 0.0201, 0.0189, ... ]
' and' ->	8 ->	[ 0.0523, -0.0208, -0.0254, ... ]
' of' ->	9 ->	[ -0.0732, 0.0070, -0.0286, ... ]
' a' ->	10 ->	[ -0.0194, 0.0302, -0.0838, ... ]
...		

시간T	M				
I	0.54	0.79	1	0	0.24
like	0.65	0.27	0.8	0.89	0.43
an	0.1	0.99	0.77	0.05	0.98
apple	0.29	0.76	0.21	0.95	0.5
and	0.55	0.95	0.29	0.22	0.44
banana	0.28	0	0.58	0.17	0.95
but	0.86	0.37	0.46	0.35	0.13
...	...	...	...	...	...



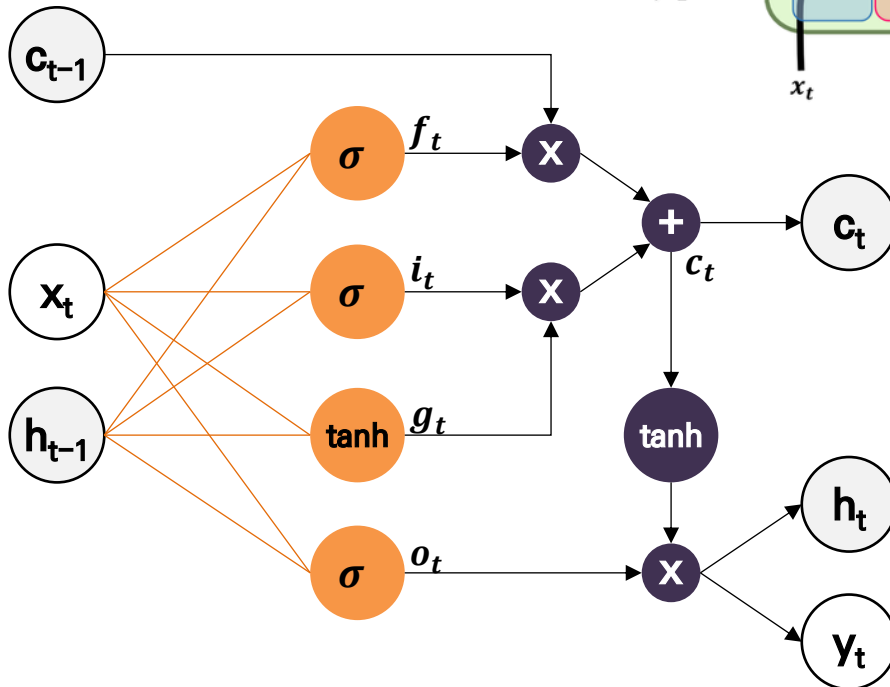
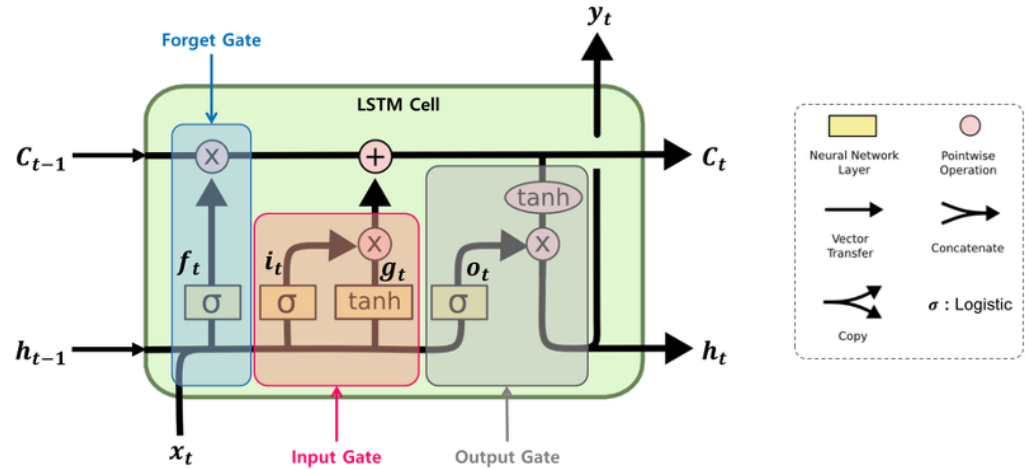
## 상태진단 모델 개발

### Base Tech

#### LSTM(Long-Short Term Memory)

##### 4개의 게이트 개념 도입

- Input, Forget, Output, Cell



$$i_t = \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{(t-1)} + b_{hi})$$

$$f_t = \sigma(W_{if}x_t + b_{if} + W_{hf}h_{(t-1)} + b_{hf})$$

$$g_t = \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{(t-1)} + b_{hg})$$

$$o_t = \sigma(W_{io}x_t + b_{io} + W_{ho}h_{(t-1)} + b_{ho})$$

$$c_t = f_t * c_{(t-1)} + i_t * g_t$$

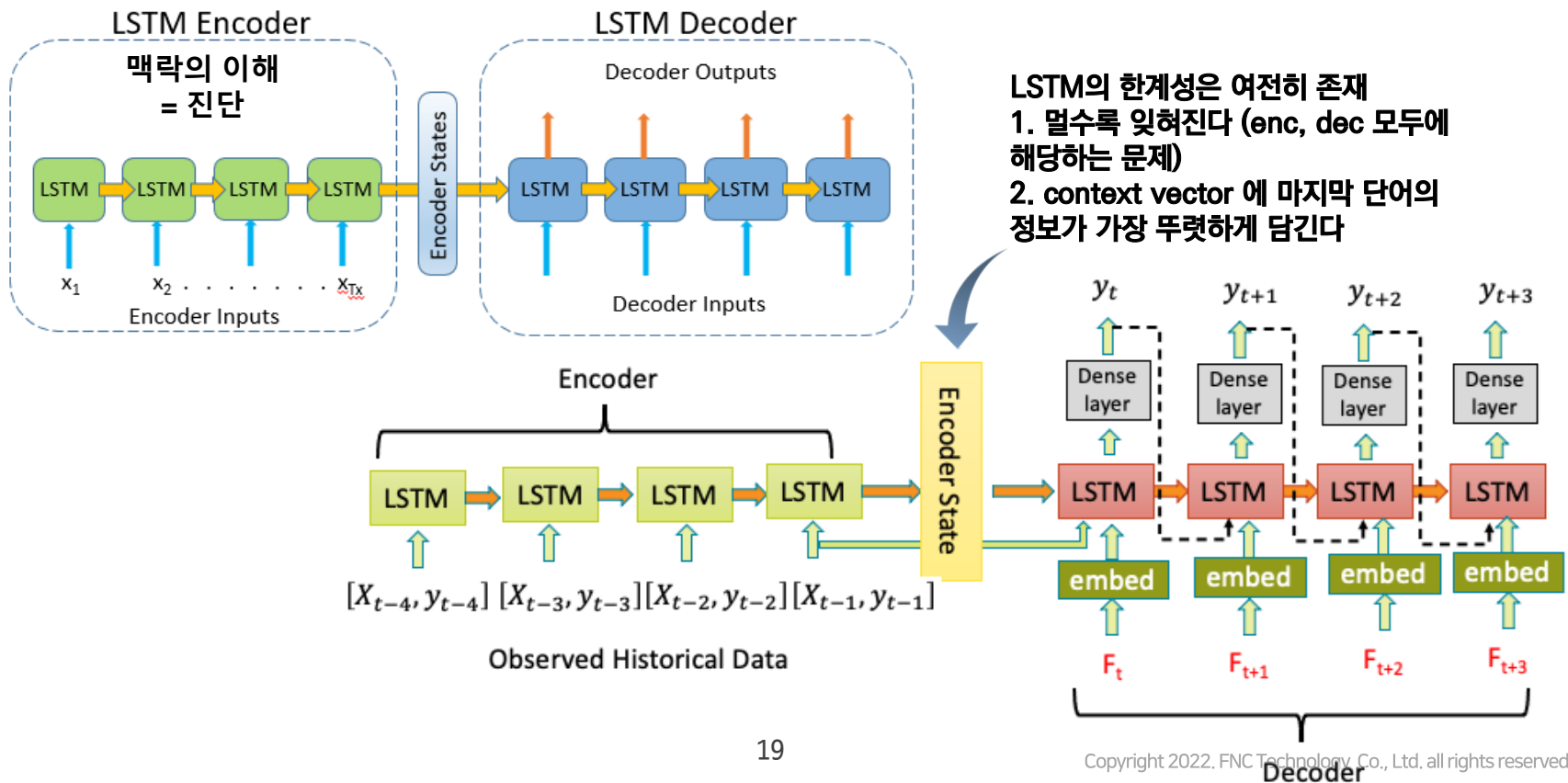
$$h_t = o_t * \tanh(c_t)$$

## 상태진단 모델 개발

### Base Tech

#### Seq2Seq

- IDEA : 입력을 벡터로 요약하는 모델을 만들
- 이후, 해당 모델이 만든 벡터를 이용해 새로운 시계열 시퀀스를 생성
- 오차 누적에 따른 장기예측의 한계성을 극복하기 위해 Long sequence를 한번에 예측



## 상태진단 모델 개발

### Base Tech

#### 왜 RNN은 번역기로는 잘 안쓰일까?

- 마지막 단어를 가장 중요시 본다? 인간의 사고 방식이 아님!
- LSTM, GRU가 있다지만..근본적인 해결은 아님!
- 이전 정보를 얼마나 끌고갈지, 현재 정보를 얼마나 담을지를 학습

#### 트랜스포머는 왜 성공했나? (논문 제목: Attention is all you need)

- Attention은 트랜스포머 이전에 이미 있던 개념
- RNN -> RNN + attention -> 트랜스포머 로 발전
- 어떤 단어를 "주목"할지를 학습한다..!

#### 트랜스포머는 attention을 적극 활용, self-attention을 통해 RNN을 완전히 버렸다..!

- 1. Decoder가 마지막 단어만 열심히 보는 문제 (갈수록 흐려진다) <- attention 으로 해결!
- 2. RNN의 vanishing gradient (멀수록 잊혀진다) <- self-attention 으로 해결!
- 3. 흐려지는 정보에 attention <- self-attention 으로 해결!

I	0.54	0.79	1	0	0.24
like	0.65	0.27	0.8	0.89	0.43
an	0.1	0.99	0.77	0.05	0.98
apple	0.29	0.76	0.21	0.95	0.5
and	0.55	0.95	0.29	0.22	0.44
banana	0.28	0	0.58	0.17	0.95
but	0.86	0.37	0.46	0.35	0.13
...	...	...	...	...	...



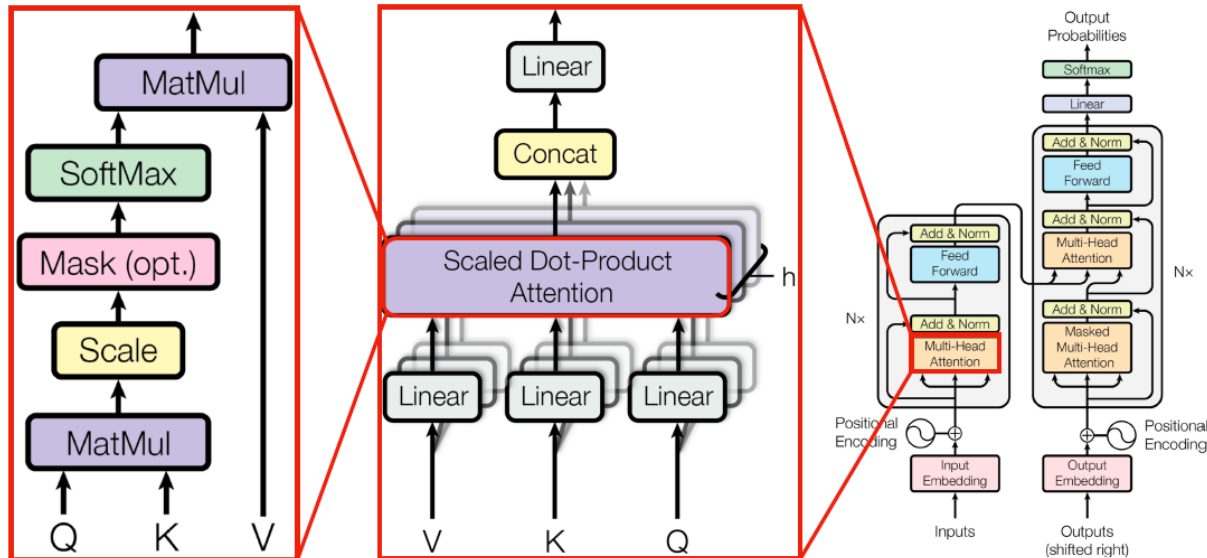
## 상태진단 모델 개발

### Base Tech

#### Attention Mechanism

- 입력 문장 내에서 단어 간의 관계를 학습시키기 위해 내적을 사용한다!
- 그런데, 어떤 단어와 어떤 단어가 관계가 깊은지 즉, 어떤 단어를 주목해야 할지를 AI가 학습해서 알아내야 할텐데.. 내적은 파라미터가 필요한 연산이 아님!! 그럼 대체 뭘 학습 파라미터로 삼는가..?
- 내적할 단어의 임베딩 벡터를 선형 변환하는 FC layer(linear)를 앞에 놓고 애를 학습시키자!
  - Query : 관계 물어볼 기준 단어 벡터
  - Key : 기준 단어와 관계를 알아볼 단어 벡터
  - Value : 관계 알아볼 단어의 의미를 담은 벡터

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



$$\vec{A} \cdot \vec{B} = |\vec{A}| |\vec{B}| \cos \theta$$

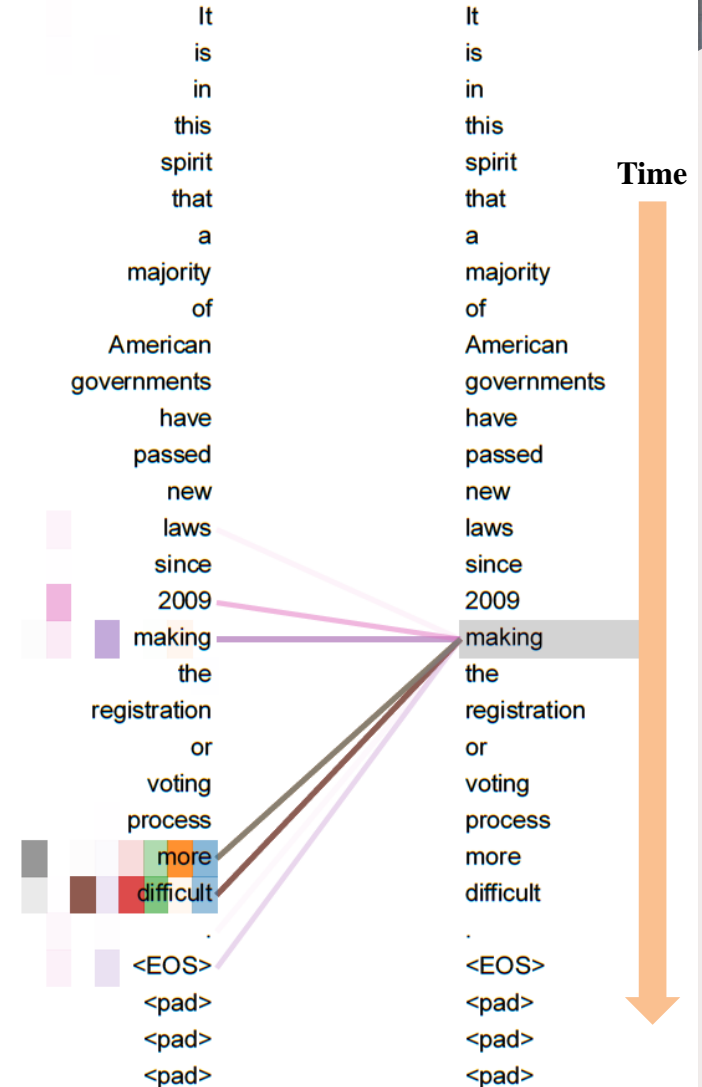
A vector diagram showing two vectors, A and B, originating from the same point. The angle between them is labeled θ. The formula  $\vec{A} \cdot \vec{B} = |\vec{A}| |\vec{B}| \cos \theta$  is shown above the vectors.

## 상태진단 모델 개발

### ➤ Base Tech

#### ■ Attention Mechanism

Time	0.248	0.202	0.055	0.445	0.919
	0.208	0.349	0.836	0.293	0.400
	0.489	0.024	0.755	0.031	0.711
	0.206	0.685	0.112	0.256	0.532
	0.057	0.541	0.644	0.680	0.258
	0.658	0.027	0.345	0.058	0.709
	0.553	0.900	0.075	0.311	0.504
	0.818	0.395	0.910	0.994	0.916
	0.821	0.191	0.839	0.501	0.517
	0.378	0.046	0.657	0.435	0.383
	0.023	0.290	0.253	0.364	0.113
	0.272	0.431	0.923	0.656	0.028
	0.519	0.853	0.099	0.344	0.868
	0.949	0.752	0.348	0.977	0.079
	0.740	0.097	0.037	0.672	0.672
	0.015	0.003	0.536	0.656	0.277
	0.965	0.855	0.669	0.445	0.514
	0.408	0.383	0.393	0.821	0.877
	0.745	0.164	0.820	0.181	0.915
	0.266	0.058	0.239	0.481	0.737
	0.019	0.618	0.399	0.038	0.118
	0.896	0.254	0.258	0.379	0.537





## 상태진단 모델 개발

### ▶ Time Series Classification 모델

#### ■ Papers with code 에서 ArabicDigits Dataset에 대한 Model Rank

Rank	Model	Accuracy↑	NLL	Paper	Code	Result	Year	Tags
1	ConvTran	0.9945		Improving Position Encoding of Transformers for Multivariate Time Series Classification			2023	
2	GP-Sig-GRU	0.994	0.023	Bayesian Learning from Sequential Data using Gaussian Processes with Signature Covariances			2020	
3	FCN-SNLST	0.993		Seq2Tens: An Efficient Representation of Sequences by Low-Rank Tensor Projections			2020	
4	GP-Sig-LSTM	0.992	0.047	Bayesian Learning from Sequential Data using Gaussian Processes with Signature Covariances			2020	LSTM
5	MALSTM-FCN	0.99		Multivariate LSTM-FCNs for Time Series Classification			2018	LSTM
6	GP-GRU	0.986	0.066	Bayesian Learning from Sequential Data using Gaussian Processes with Signature Covariances			2020	

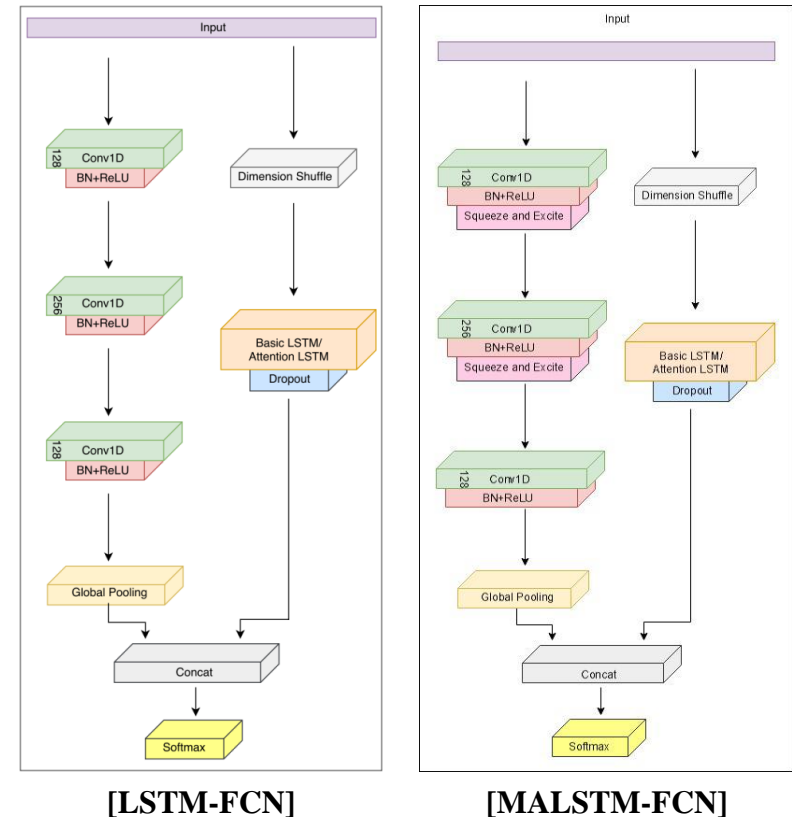
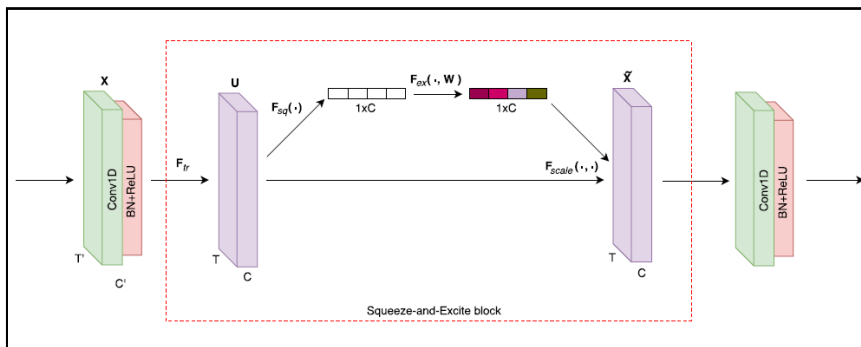
#### ■ ConvTran, MALSTM-FCN 활용

- GP-Sig-LSTM, GP-Sig-GRU 는 Bayesian Learning 을 통한 ML 기술(Gaussian process (GP), Signature Covariance)
- FCN-SNLST : Seq2Tens: An Efficient Representation of Sequences by Low-Rank Tensor Projections 논문 확인 예정

## 상태진단 모델 개발

### Time Series Classification 모델

- LSTM Fully Convolutional Networks for Time Series Classification, September 2017 IEEE Access PP(99)
  - Convolutional blocks
  - Batch normalization
  - ReLU activation
  - Attention LSTM : be used detect regions of the input sequence that contribute to the class label
- Multivariate Attention LSTM-FCN : *Multivariate LSTM-FCNs for Time Series Classification*, Neural Networks, Volume 116, August 2019
  - Squeeze-and-Excite block(S&E Block)



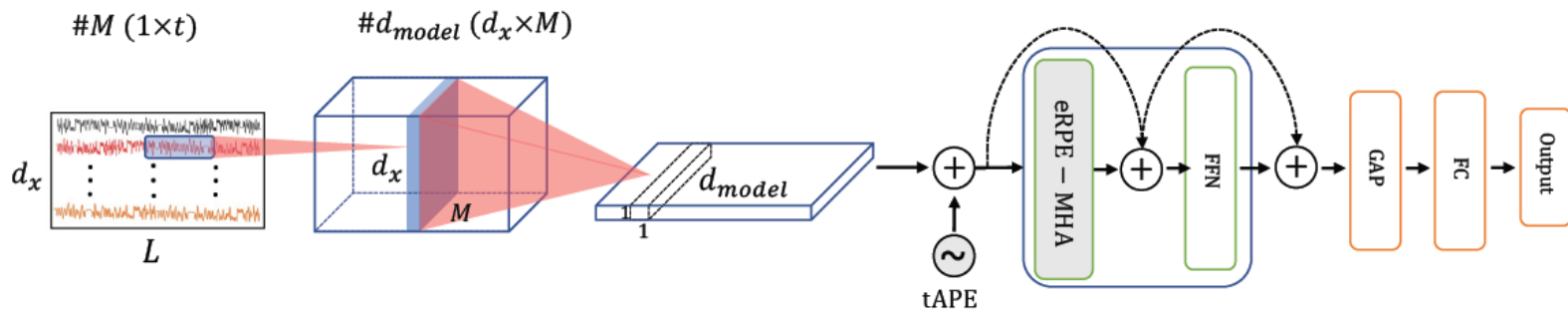
### 삼진분류 테스트 결과

- Validation Dataset Accuracy : 44.92 %
- Test Dataset Accuracy : 40.18 %

## 상태진단 모델 개발

### ▶ Time Series Classification 모델

- ConvTran : *Improving Position Encoding of Transformers for Multivariate Time Series Classification*, Springer Nature, 05 September 2023



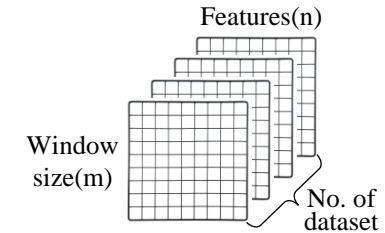
- CNN with  $\#M$  filter : extracts temporal patterns in the input series
- CNN with  $\#d_{model}$  filter : capture the correlations between variables  $\rightarrow d_{model}$  size input embeddings
- Add Time Absolute Position Encoding (tAPE) : capture the temporal order of the time series( $L \times d_{model}$ )
- Multi-Head Attention block with Efficient Relative Position Encoding (eRPE)
  - 중요한 정보 포착 : 중요한 특징에 더 많은 가중치를 부여하고, 덜 중요한 정보는 무시
  - 시간적 의존성 학습 : 시간적 의존성을 학습하며, 시계열의 다양한 시점 간의 복잡한 관계를 이해
  - 긴 범위 의존성 해결 : 긴 시퀀스에서 발생할 수 있는 긴 범위 의존성 문제를 효과적으로 해결
  - 병렬 처리 : 시퀀스의 모든 요소를 한 번에 처리할 수 있어, 모델의 학습과 추론 과정에서 높은 병렬 처리 능력 제공

## 상태진단 모델 개발

### ▶ 학습 데이터셋

	PPS	TCRHOT	MH2CR1	ZWV	PPSTRB(3)	MFPWB(2,3)	WSPTA
TIME	features						
0.000000	1.553135e+07	988.374430	0.000000	6.712370	10497.684600	0.000000	0.000000
0.001000	1.553164e+07	988.374430	0.000000	6.712370	10497.686728	0.000000	0.000000
300.098525	1.374299e+07	589.749630	0.000000	6.712370	10610.922523	0.000000	0.000000
600.098525	1.330853e+07	584.539460	0.000000	6.712370	10637.030370	0.000000	0.000000
900.098525	1.309044e+07	580.881991	0.000000	6.712370	10647.300461	0.000000	0.000000
...	...	...	...	...	...	...	...
321487.720099	2.783103e+05	100.000000	108.330616	4.312400	21266.333103	0.000002	171.480895
321788.537966	2.781354e+05	100.000000	108.330616	4.535665	21184.360080	0.000002	171.497192
322089.382649	2.779998e+05	100.000000	108.330616	4.309454	21103.305550	0.000002	171.513331
322390.074890	2.778261e+05	100.000000	108.330616	4.520843	21037.116768	0.000003	171.529308
322630.432005	2.777315e+05	100.000000	108.330616	4.302035	20976.149013	0.000002	171.541975

1076 rows × 14 columns



Ex)

No. of data row = 1076

No. of feature(n) = 14

Window size(m) = 4

→ No. of dataset(m by n)  
= 1076-4+1 = 1073

### ▶ 분류모델의 성능기준

#### Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$F1 \text{ Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

- Precision이 높다는 것은 거짓 양성(FP)의 수가 적다는 것을 의미
- Recall이 높다는 것은 거짓 음성(FN)의 수가 적다는 것을 의미
- 양성과 음성을 정확하게 분류하는 것이 모두 중요한 경우 F1-score

## 과제 수행 내용

### 상태진단 모델 개발

#### ▶ 여덟 가지 레이블, Window size 12

■ Learning rate 2e-4, epoch 5,000, embedding size 64, ff dimension 512, dropout 0.1

■ Trainable Parameters : 416,626 (약 3400 KB)

■ Learning rate 2e-4, epoch 5,000, embedding size 32, ff dimension 256, dropout 0.2

■ Trainable Parameters : 110,450 (약 940 KB)

True   Pred	Precision	Recall	F1-score	# of test data
0	0.903	0.814	0.856	10,427
1	0.940	0.935	0.937	8,232
2	0.998	1.000	0.999	16,287
3	0.931	0.938	0.934	7,737
4	0.999	1.000	0.999	12,424
5	0.999	0.995	0.997	12,205
6	1.000	1.000	1.000	8,903
7	0.832	0.914	0.871	10,417
Average	0.956	0.955	<u>0.955</u>	86,632

#### [최적 Fine-Tuning 결과]

- 학습 소요시간 약 39 시간 40 분
- 4795 epoch에서 Validation Accuracy 최대

True   Pred	Precision	Recall	F1-score	# of test data
0	0.882	0.769	0.822	10,427
1	0.910	0.894	0.902	8,232
2	0.998	0.999	0.999	16,287
3	0.893	0.911	0.902	7,737
4	0.999	1.000	1.000	12,424
5	0.995	0.995	0.995	12,205
6	1.000	0.999	0.999	8,903
7	0.799	0.899	0.846	10,417
Average	0.942	0.941	<u>0.941</u>	86,632

#### [경량화 Fine-Tuning 결과]

- 학습 소요시간 약 22 시간 6 분
- 4695 epoch에서 Validation Accuracy 최대



## 결과예측 모델 개발

### Time-Series Forecasting(TSF) 방법론

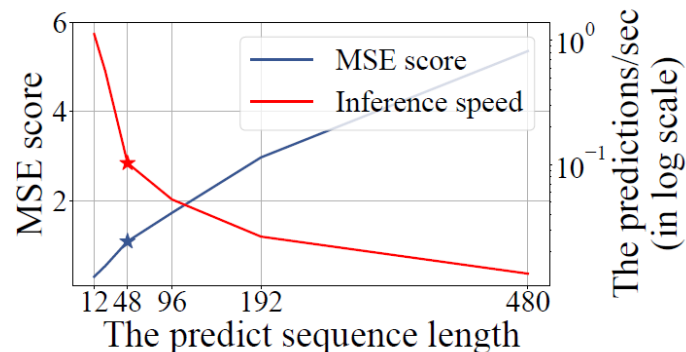
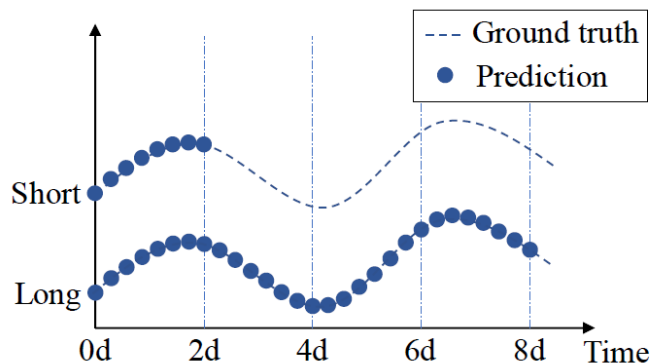
- RNN (1986) : RNN 첫 시작. 10년 후 LSTM
- LSTM (1997) : 다양한 시퀀스 정보 모델링 가능 - 주가예측, 주기함수 예측
- Seq2Seq (NIPS 2014)
  - LSTM 활용해서 딥러닝 기반 기술로 탄생. 현대의 딥러닝 기술들이 나오던 시점에 탄생.
  - LSTM 을 활용해서 고정된 크기의 context vector 를 사용하는 방식으로 번역 수행.
  - 따라서 소스 문장을 전부 고정된 크기의 한 벡터에 압축해야 하는 성능적인 한계가 존재했음(bottleneck)
- Attention (ICLR 2015) : Seq2Seq 모델에 어텐션 기법 적용하여 성능 더 올릴 수 있었음
- Transformer (NIPS 2017) 'Attention is all you need.'
  - 아예 RNN 자체를 사용할 필요가 없다고 제안.
  - 오직 어텐션 기법에 의존하는 아키텍처를 설계했더니 성능이 엄청 향상.
  - 트랜스포머를 기점으로 자연어 처리 기법으로 RNN 더 이상 사용하지 않고 어텐션 메커니즘을 더욱 더 많이 사용
  - 어텐션 메커니즘 등장 이후 입력 시퀀스 전체에서 정보를 추출하는 방향으로 연구 방향 발전.
  - 어텐션 기법을 활용하는 트랜스포머 아키텍처를 따르는 방식으로 다양한 고성능 모델들이 제안되고 있음.



## 결과예측 모델 개발

### ➤ RNN + Attention 적용의 한계성

- 병렬화 문제 : RNN은 그 구조상 순차적으로 입력을 처리해야 하기에 병렬화가 불가능하다. 이 때문에 대규모의 데이터셋을 이용한 학습이 불가능함(학습 시간이 너무 길어진다).
- long distance dependency 문제 : 시퀀스에서 멀리 떨어진 항목들 간의 관계성은 gradient vanishing/exploding 문제로 학습이 잘 되지 않음.
- The prediction capacity of RNN+Attention methods limits long-term prediction's performance. From length=48, MSE rises unacceptably high, and the inference speed drops rapidly.



### ➤ Transformer(2017) limitations when solving the Long-term problem

- 메모리 사용량 : 시퀀스 길이가 길어질수록 계산량과 메모리 사용량이 기하급수적으로 증가
- 실제로 긴 시계열 데이터에서는 중요한 정보의 위치를 예측하기 어려움
- 긴 시계열에서는 포지셔널 인코딩이 효과적이지 않을 수 있으며, 시간의 흐름에 따른 패턴 학습 한계

## 결과예측 모델 개발

## eQRNN 모델

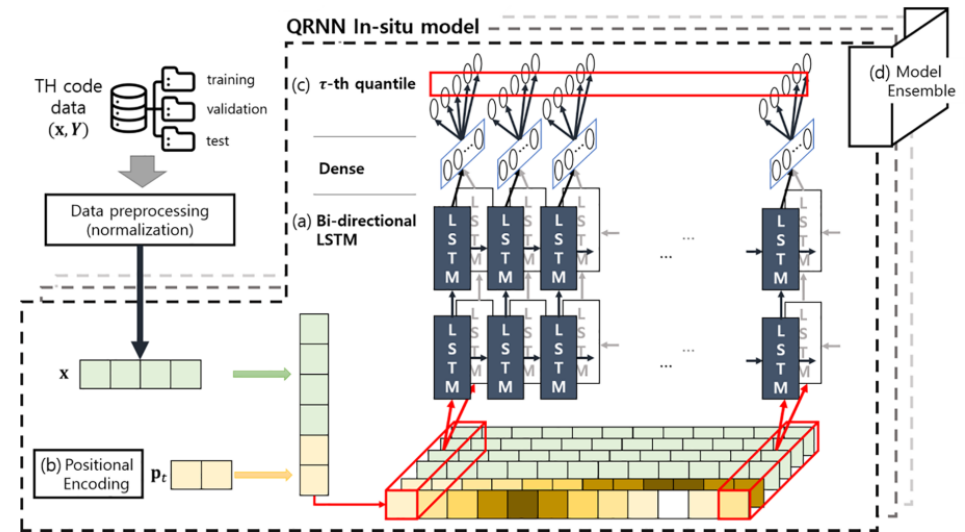
## 주요 기술요소

- Positional Encoding(PE)
- Bi-directional LSTM
- Quantile regression
- Model Ensemble

## Input(X) : constant vector

- X1~X16 concat PE1~PE8

## Output(Y) : Target time-series



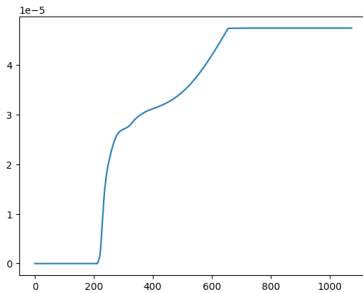
	Y	X1	X2	...	X16	PE1	PE2	...	PE8
T1	0.2929	0.58	0.12	...	0.91	0	1	...	1
T2	0.1531	0.58	0.12	...	0.91	0.8414	0.5403	...	1
T3	0.2220	0.58	0.12	...	0.91	0.9092	-0.4161	...	0.9999
T4	0.2221	0.58	0.12	...	0.91	0.1411	-0.9899	...	0.9999
⋮	⋮	⋮	⋮	...	⋮	⋮	⋮	...	⋮
T358	0.7747	0.58	0.12	...	0.91	-0.9092	0.4161	...	0.9369
T359	0.7740	0.58	0.12	...	0.91	-0.1410	0.9899	...	0.9365
T360	0.7163	0.58	0.12	...	0.91	0.7568	0.6536	...	0.9362

Probabilistic deep learning model as a tool for supporting the fast simulation of a thermal hydraulic code, S.Ryu et al., Expert Systems with Applications 200, 2022

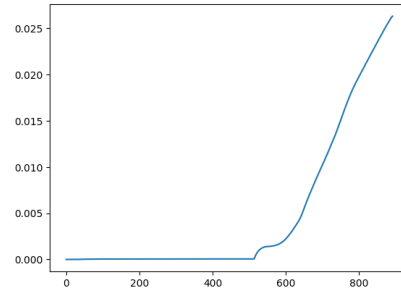
## 결과예측 모델 개발

## ▶ eQRNN Test : 격납건물 내 Csl 질량분률 예측

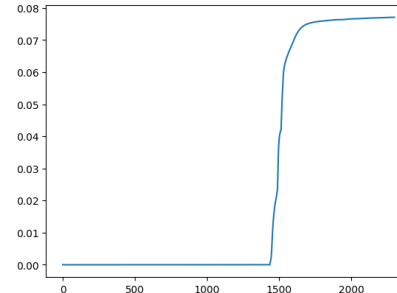
## ■ Target Sequence 별 고정 벡터 (1 x 16 shape)



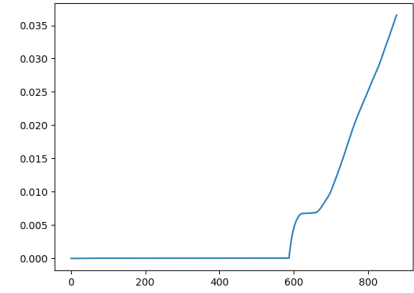
[[0.94, 0.87, 0.69, 0.74,  
0.16, 0.21, 0.86, 0.56,  
0.17, 0.19, 0.69, 0.54,  
0.15, 0.82, 0.20, 0.15]]



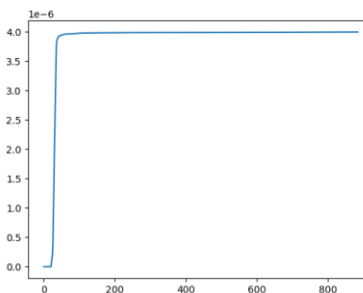
[[0.12, 0.35, 0.28, 0.10,  
0.88, 0.14, 0.70, 0.74,  
0.26, 0.47, 0.93, 0.23,  
0.68, 0.64, 0.79, 0.91]]



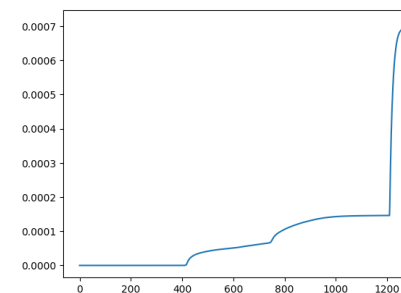
[[0.96, 0.99, 0.79, 0.86,  
0.61, 0.94, 0.69, 0.06,  
0.25, 0.14, 0.99, 0.17,  
0.38, 0.46, 0.50, 0.01]]



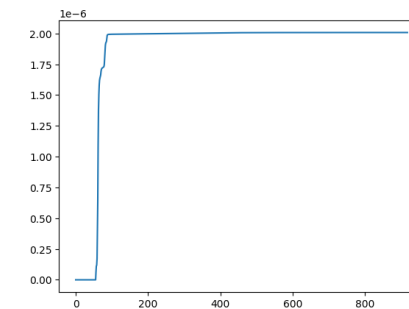
[[0.58, 0.12, 0.91, 0.60,  
0.06, 0.46, 0.23, 0.69,  
0.10, 0.03, 0.90, 0.37,  
0.43, 0.34, 0.25, 0.91]]



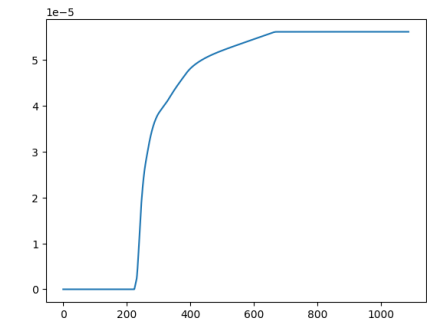
[[0.33, 0.56, 0.02, 0.73,  
0.34, 0.99, 0.84, 0.82,  
0.72, 0.62, 0.18, 0.73,  
0.98, 0.37, 0.69, 0.28]]



[[0.95, 0.07, 0.86, 0.35,  
0.93, 0.52, 0.43, 0.83,  
0.86, 0.52, 0.63, 0.04,  
0.12, 0.70, 0.52, 0.07]]



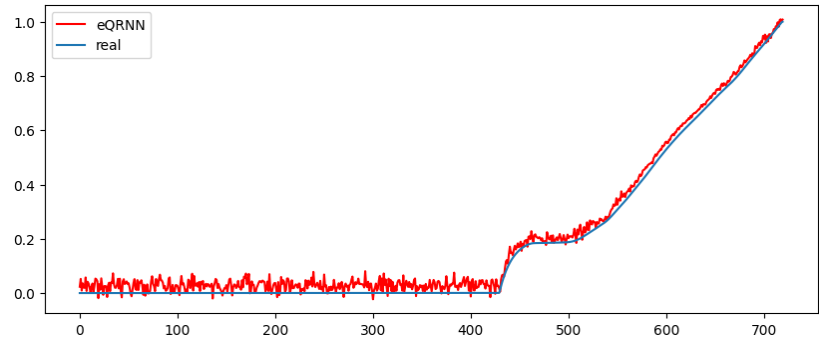
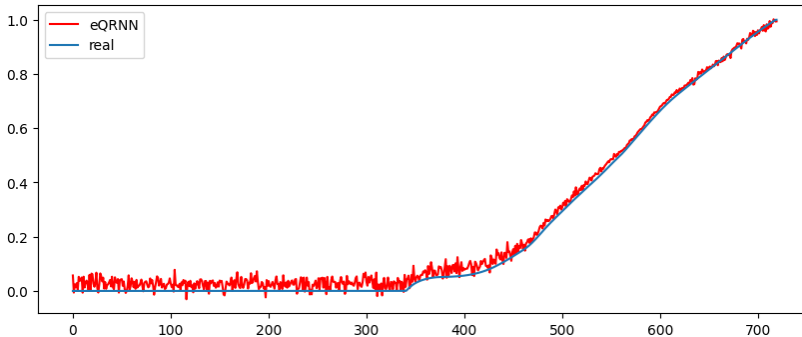
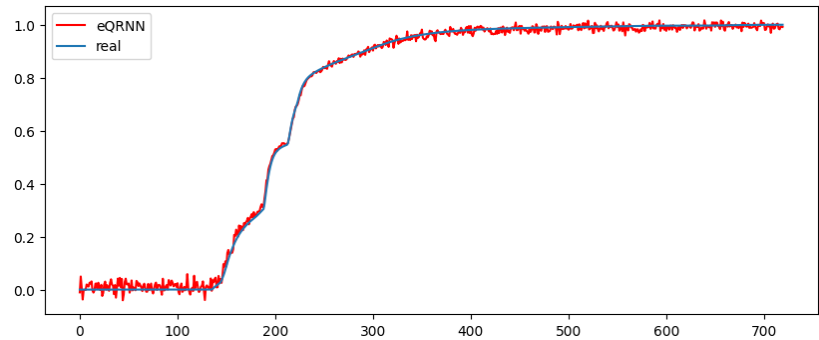
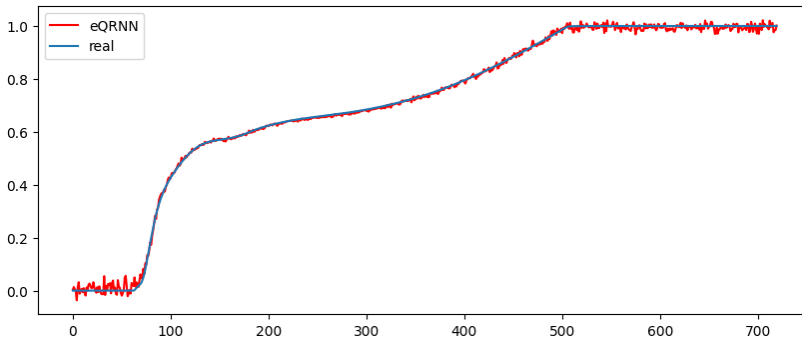
[[0.77, 0.19, 0.67, 0.96,  
0.15, 0.83, 0.62, 0.20,  
0.10, 0.65, 0.76, 0.49,  
0.08, 0.04, 0.64, 0.55]]



[[0.07, 0.91, 0.02, 0.64,  
0.20, 0.32, 0.20, 0.02,  
0.16, 0.47, 0.14, 0.24,  
0.95, 0.74, 0.13, 0.05]]

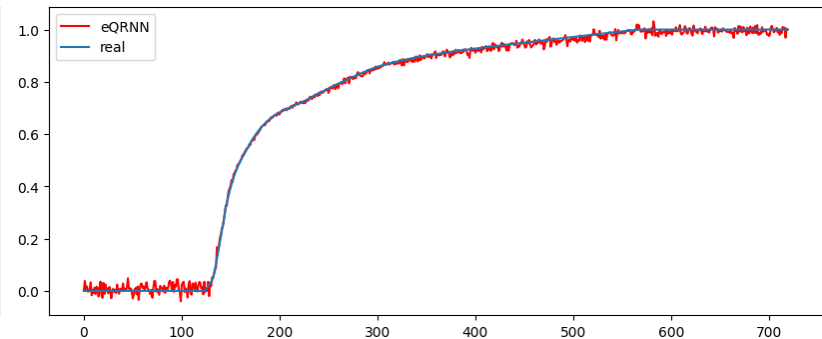
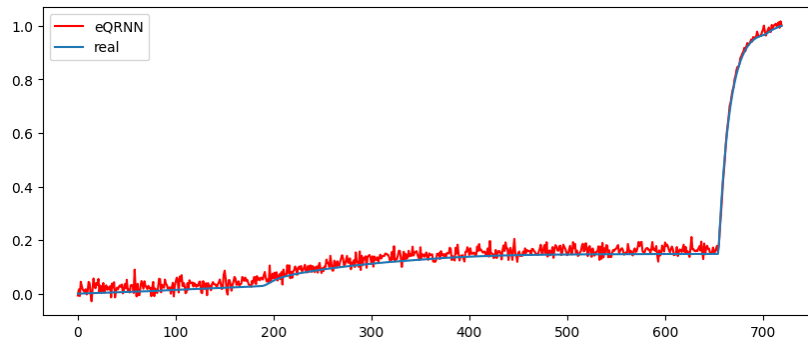
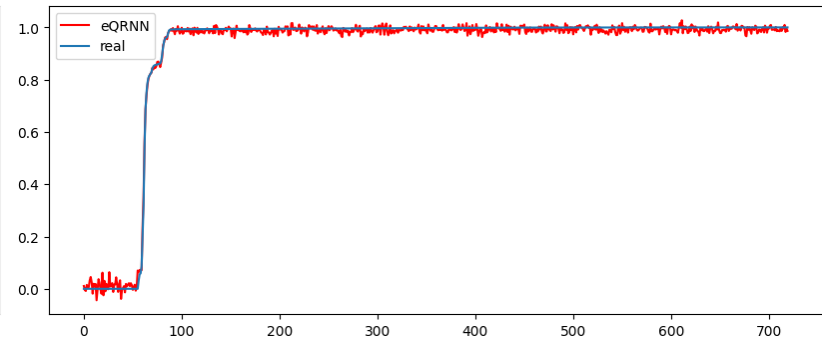
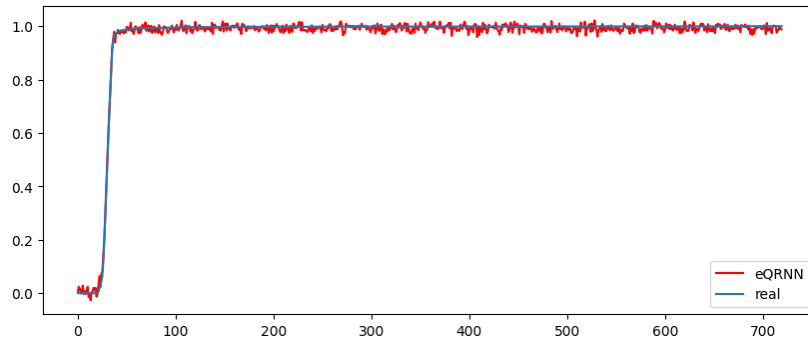
## 결과예측 모델 개발

## ➤ eQRNN Test : 격납건물 내 CsI 질량분률 예측



## 결과예측 모델 개발

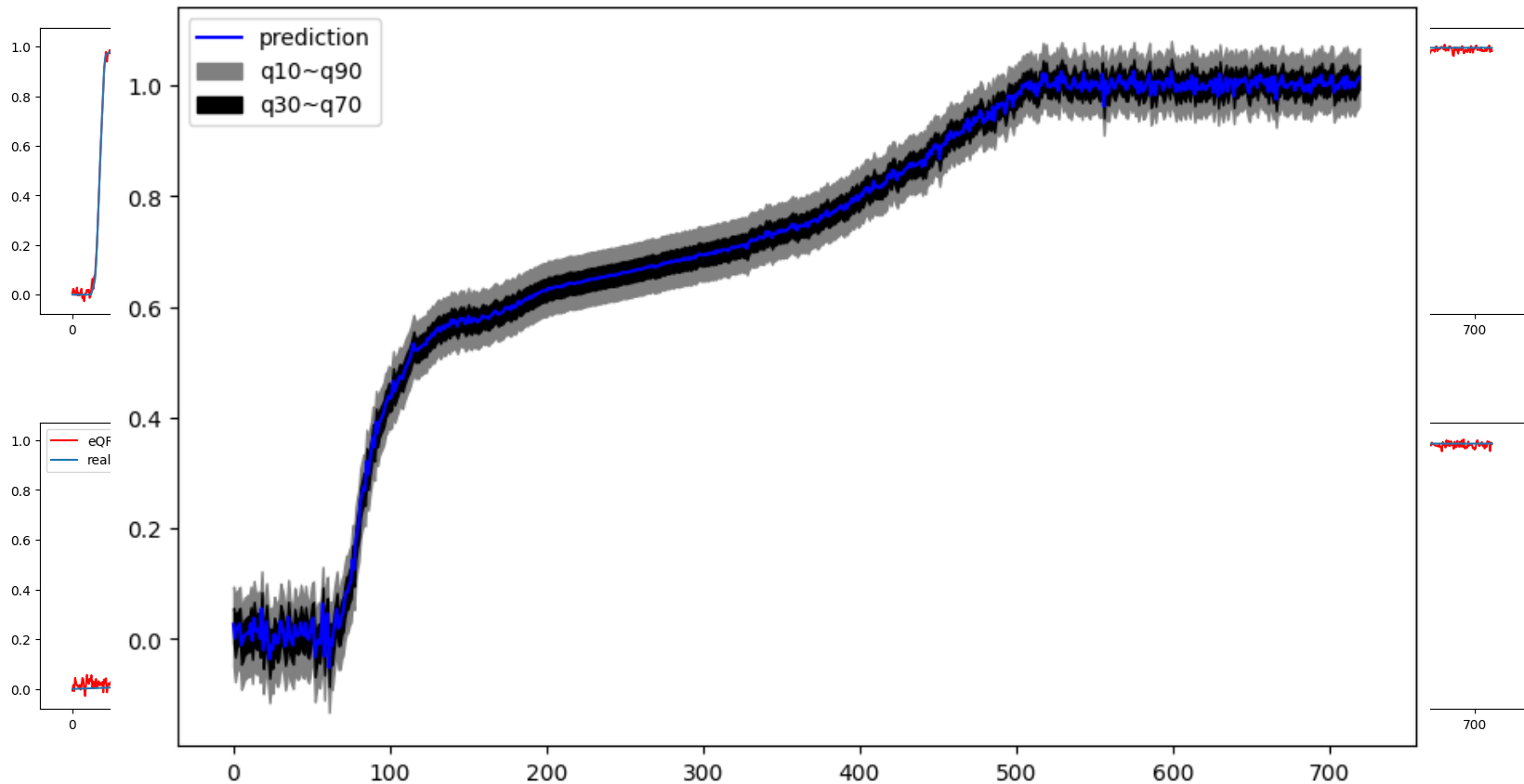
## ➤ eQRNN Test : 격납건물 내 Csl 질량분률 예측





## 결과예측 모델 개발

## ▶ eQRNN Test : 격납건물 내 Csl 질량분률 예측



# 4

## Considerations

## 딥러닝 적용성 및 활용 가능성

### › (3-1-1-산) '24년 과제 연구 계획

#### ■ 고려중인 초기사건 : LLOCA, MLOCA, SLOCA, SGTR, SBO, LOFW

	SA 진입전 Event											
<b>LLOCA</b>	SIT	SIS	HIN	CSR								
<b>MLOCA</b>	SIS	HIN	CSR									
<b>SLOCA</b>	SIS	SHR	ASC	SCSI	SDC	MSHR	SD	SIF	CHR	CSR		
<b>SGTR</b>	SIS	ASC1	SHR	ISLK	SCSI	SDC	RF	MSHR	SD	SIF	CHR	CSR
<b>SBO</b>	AAC	AFWT	RSF	RACE	SHR	RACL	SDC	MSHR	SD	SIF	CHR	CSR
<b>LOFW</b>	SHR	SDC	MSHR	SD	SIF	CHR	CSR					

#### ■ 고려중인 완화전략 : SADP, SASI, SACF, SACS, SAFC 등

#### ■ 시나리오 수 : 약 50,000개

#### ■ 올 한해 데이터 확보와 Prototype 서비스 개발에 초점

### › 중대사고 데이터의 딥러닝 기술 적용성

- 중대사고 데이터는 Multivariate Time-Series로써 각 time-step별 생산되는 데이터가 자연어 처리에서 활용되는 Word embedding Vector와 동일함.
- 현재 가장 활발하게 진화중인 자연어처리 분야의 딥러닝 기술들을 중대사고 데이터에 적용하여 중대사고의 진단 및 발전소 거동 예측이 가능함.
- 다만, 제한적인 사고환경에서의 적용성 확인보다 일반적인 환경에서 적용성 확인을 위해 데이터 확장 및 모델구조 개선이 필요함.
- 추가적으로 발전소 내 시스템 구축을 위한 다양한 필요조건도 고민 필요(설명가능성, V&V, 시스템환경 등)

# 4 Considerations

## For Next One Step

- 모든 산업에서 인공지능 기술 적용은 선택이 아닌 필수의 시대로 가고 있음
- 다년간의 연구 결과로 원자력 분야에 딥러닝 기술 적용이 가능하다는 것을 확인했음
- 딥러닝 모델 개발을 위해서 필요한 것은 '데이터가 전부다' 라고 할 수 있을 정도로 기술 문턱은 낮아짐
- 중대사고 데이터에 대한 여러분의 생각은?
  - 실제 중대사고 데이터가 존재하는가?
    - TMI, 체르노빌, 후쿠시마 데이터는 우리 원전에 유효한가?
    - 유일한 데이터는 코드해석 데이터가 전부
  - 그렇다면, 해석 데이터는 잘 보존하고 있는가?
    - 개별 PC에서 저장되다가 디스크/보안 문제로 삭제
  - 데이터의 가치에 공감하고 있는가?
    - 맞는지 틀린지도 모르는 데이터를 왜 모으냐?
- 해석 업무의 데이터 중심 디지털전환 필요
  - 산업디지털전환촉진법 - 2022.7.5 시행
    - 정보화촉진기본법 - 1996.1.1 시행
    - 산업데이터 활용 및 보호 원칙, 선도사업 지원 등 정부 종합지원 근거
  - Digitization → Digitalization → Digital Transformation(일하는 방식의 변화)
- 중대사고 해석 분야의 디지털 전환을 통해 데이터 중심의 해석체계 구축 필요

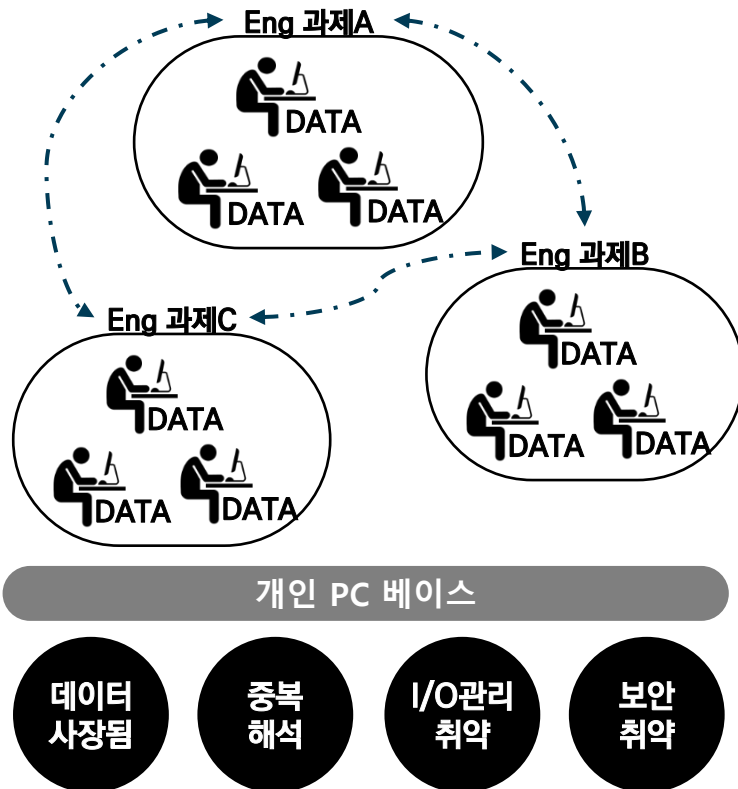


# 4 Considerations

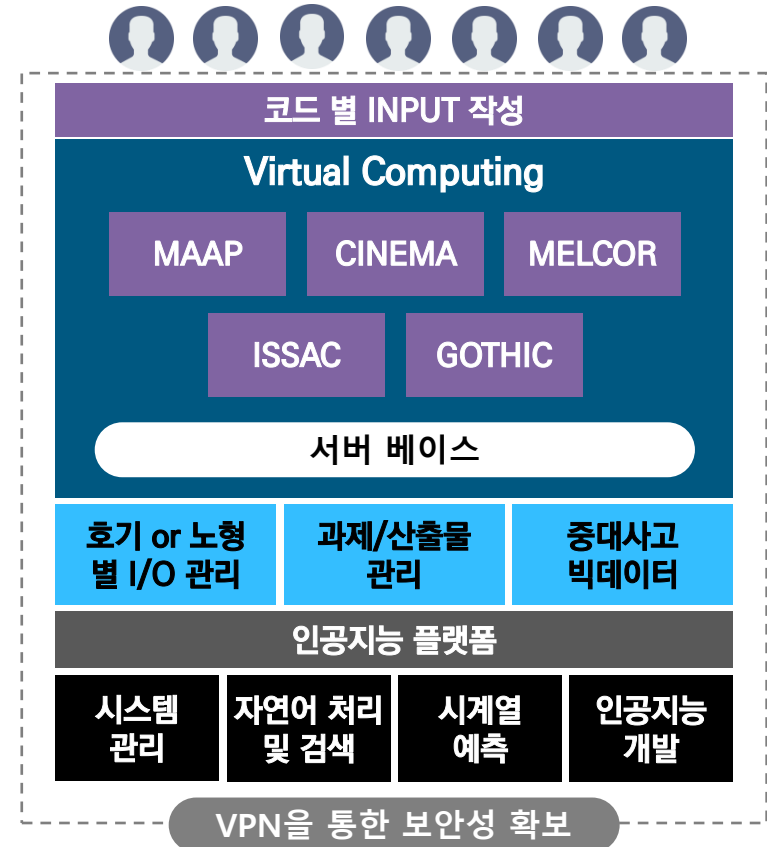
## For Next One Step

### 중대사고 해석 분야의 디지털 전환(DX)

#### As-Is



#### To-Be



# ABOUT



**|주|미래와도전**  
FNC Technology Co., Ltd.

## 📍 미래와도전 본사

16954 경기도 용인시 기흥구 흥덕1로 13, 32층  
(영덕동, 흥덕아이티밸리 타워동)  
[+82-31-8065-5114](tel:+82-31-8065-5114)

## 📍 미래에너지기술연구소 본관

경기도 용인시 기흥구 탑실로 46,  
미래에너지기술연구소 본관  
[+82-31-8005-5618](tel:+82-31-8005-5618)

## 📍 UAE 아부다비지사

#2335, Sky Tower, Al Reem Island PO Box 5101041,  
Abu Dhabi, UAE  
[+971-2-406-9719](tel:+971-2-406-9719)

## 📍 대전지사

대전광역시 유성구 대덕대로 593, 10층 1004-1호  
(도룡동, 대덕테크비즈센터)  
[+82-42-867-5114](tel:+82-42-867-5114)

## 📍 미래에너지기술연구소 신관

경기도 용인시 기흥구 탑실로 44,  
미래에너지기술연구소 신관  
[+82-31-8005-8236](tel:+82-31-8005-8236)

## 📍 미국 메릴랜드지사

11140 Rockville Pike, Suite 380A,  
Rockville, MD 20852 USA  
[+1-202-577-5537](tel:+1-202-577-5537)



# THANK YOU



FNC TECHNOLOGY CO., LTD.