

한국 신뢰성 학회-현대자동차 퀴즈 세션

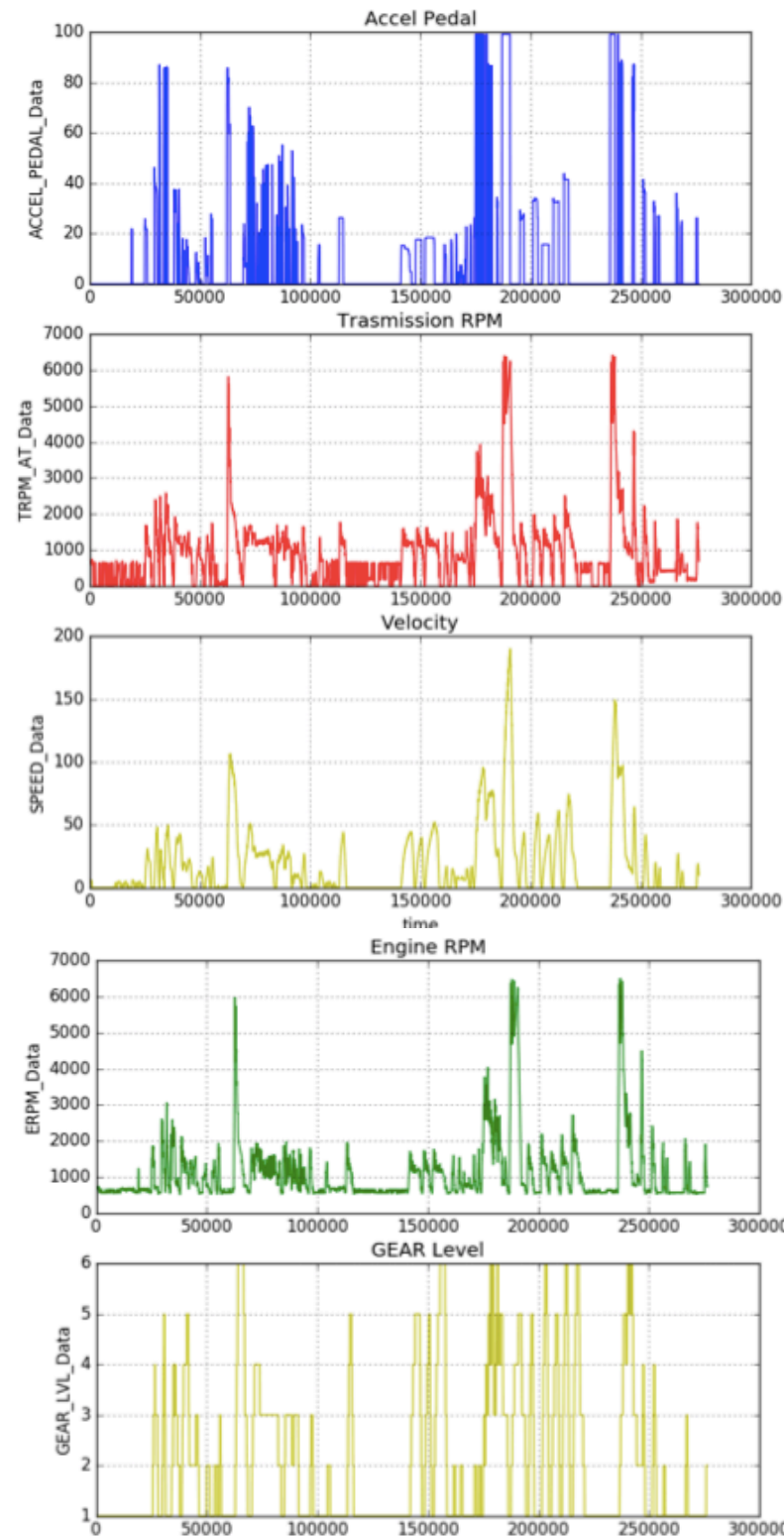
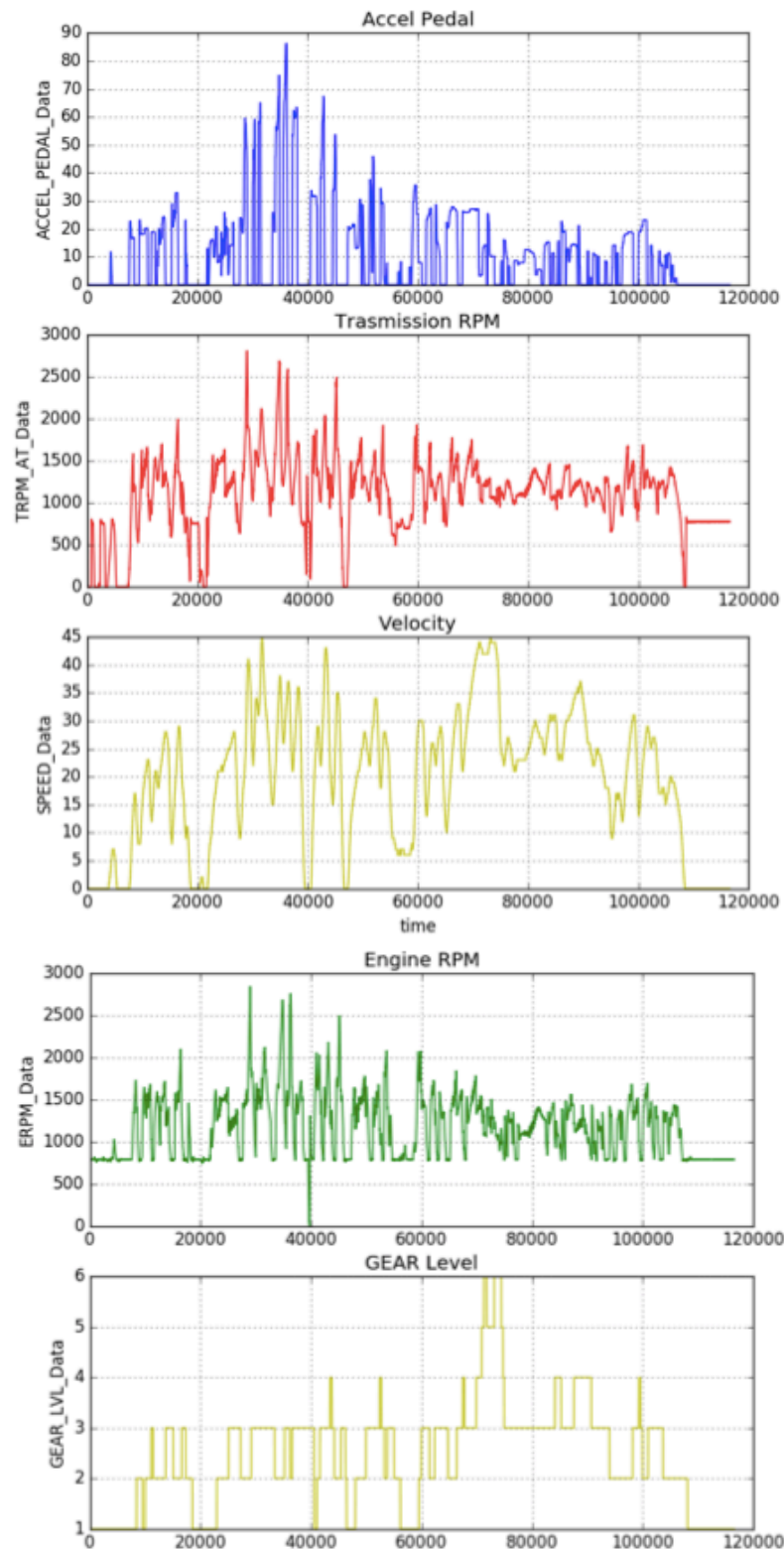
# Machine Learning 과 SPC를 이용한 차량 센서 데이터의 비정상 구간 탐색 및 분류

연세대학교 산업공학과

고정욱 김영빈 유애리 이한별 정구봉  
지도교수: 손소영 교수님



# 연구 배경 및 목적



- ▶ raw data: 2개의 신호 데이터 셋  
 problem\_signal\_1.csv → Signal 1  
 problem\_signal\_2.csv → Signal 2
- ▶ 수집 변수 채널 (5개 채널), 분석 윈도우 (300 프레임), 수집 시간 단위 (0.02초)
- ▶ problem set 1: 116594 /problem set 2: 276327

변수명	설명
시간	Time Frame
악셀패달	악셀 패달 개도
엔진 RPM(*)	엔진 회전수
변속기 RPM(*)	변속기 회전수
기어단수	현재 기어 단수
속도	속도

# 문헌 고찰

## 다중 정상 하에서 단일 클래스 분류기법을 이용한 이상치 탐지 - TFT-LCD 공정 사례

주태우, 김성범(2013). 대한산업공학회지, 39(2), 82-89.

본 연구에서는 다중 정상 하에서 이상치 탐지 기법을 제안하였다. 특히, AND 게이트를 이용한 이상치 탐지법은 여러 개의 input을 받아 필요한 출력 값을 만드는 방법인데 다양한 이상치를 탐지하여 병렬로 혼합하는 방향을 생각하는데 참고 되었다.

## 센서 데이터 분류에 대한 분류 알고리즘 성능 비교

남윤진, 신동일, 신동규(2015). 한국정보과학회 학술발표논문집, 1891-1893.

본 연구에서는 Naïve Bayesian과 SVM을 적용하여 결과를 성능 면에서 비교하여 센서 데이터에 대한 효과적인 분류기를 선별해 내는 것을 목적으로 하였다. 우리 연구의 초기 분류 알고리즘 선택 시점에서 본 연구가 도움 되었다.

## 자동차 엔진 고장 감지 기법 비교 연구

백수정, 김덕영(2013). 한국 CDE학회 학술발표회 논문집, 243-248.

본 연구에서는 여러 개의 센서로 구성된 시스템에서 정보의 융합을 통한 분석결과를 이용해 고장을 감지하였다. 융합된 센서 결과를 Hotelling's  $T^2$  등의 관리도 기법을 통해 분석하는 부분에서 우리 연구의 센서 이상을 감지에 관리도를 적용하는 방법에 참고 되었다.

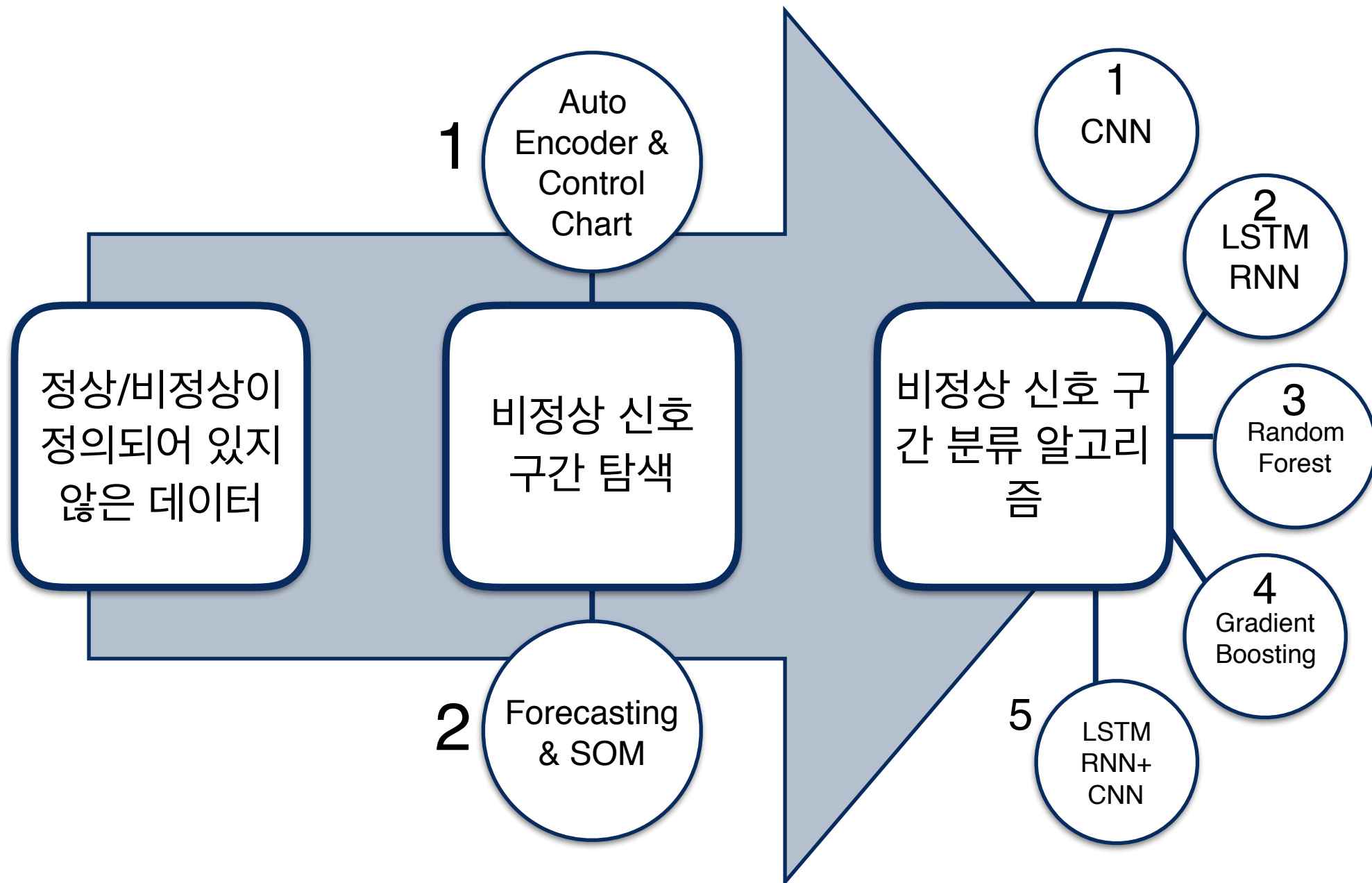
## 규칙 기반 및 상관 분석 방법을 이용한 시계열 계측 데이터의 이상치 판정

전제성, 구자갑, 박창목(2015). 한국지반환경공학회 논문집, 16(5), 43-53. 1891-1893.

본 연구에서는 형태별로 각기 다른 방식의 이상치 판정을 하는 규칙기반 판정법과 회귀분석 모델에 기반한 상관분석 판정법을 사용하였다. 특히, 규칙기반 판정법에서 회귀를 통해 오류를 예측한다는 관점에서 참고 되었다.



# 개요



# 데이터

## 데이터 전처리

- ▶ 0.02초 단위로 측정된 시간별 센서 데이터들을 **300프레임(6초)** 단위로 가공하여 분석
- ▶ **Normalized**: 각 변수의 값의 범위가 다르기 때문에 같은 기준으로 분석하기 위해 normalize
- ▶ 300\*5 짜리 데이터를 계산에 용이하게 하기 위해 1\*1500 형태로 만든 것을 기본으로 사용함

	A	B	C	D	E	F	G	H	I	J	K	L
1	Accel	Engine_RPM	ChangeGear_RPM	Gear_Level	Velocity	Accel	Engine_RPM	ChangeGear_RPM	Gear_Level	Velocity	Accel	Engine_RPM
2	-0.694001776	-1.046917484	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.046917484	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.047112824
3	-0.694001776	-1.051816861	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.053447216	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.055077572
4	-0.694001776	-1.041368729	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.041634356	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.04328377
5	-0.694001776	-1.034446199	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.037819509	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.041138495
6	-0.694001776	-1.033219629	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.036522416	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.038594352
7	-0.694001776	-1.030911553	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.030772985	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.031606309
8	-0.694001776	-1.03923593	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.038594352	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.038594352
9	-0.694001776	-1.040577503	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.038966871	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.038594352
10	-0.694001776	-1.03845702	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.036809746	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.035162472
11	-0.694001776	-1.044143106	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.044143106	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.044143106
12	-0.694001776	-1.043525975	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.044143106	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.044143106
13	-0.694001776	-1.035819975	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.035819975	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.035819975
14	-0.694001776	-1.041368729	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.041368729	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.041368729
15	-0.694001776	-1.045270768	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.046096176	-2.474547681	-1.328303818	-1.581298794	-0.694001776	-1.046921593
16	-0.694001776	-1.027496844	-1.393873911	-1.328303818	-1.581298794	-0.694001776	-1.027721168	-1.36704253	-1.328303818	-1.581298794	-0.694001776	-1.028547732
17	-0.694001776	-0.961361524	-0.659335028	-1.328303818	-1.581298794	-0.694001776	-0.962187923	-0.65594216	-1.328303818	-1.581298794	-0.694001776	-0.963014323
18	-0.694001776	-0.980565924	-0.672699748	-1.328303818	-1.581298794	-0.694001776	-0.983064004	-0.674067231	-1.328303818	-1.581298794	-0.694001776	-0.985562084
19	-0.694001776	-1.00252745	-0.7158558	-1.328303818	-1.581298794	-0.694001776	-1.003711029	-0.7158558	-1.328303818	-1.581298794	-0.694001776	-1.005360195
20	-0.694001776	-1.016399336	-0.731802488	-1.328303818	-1.581298794	-0.694001776	-1.017400237	-0.731802488	-1.328303818	-1.581298794	-0.694001776	-1.020682666
21	-0.694001776	-1.030271221	-0.741687332	-1.328303818	-1.581298794	-0.694001776	-1.030271221	-0.743041294	-1.328303818	-1.581298794	-0.694001776	-1.031003133
22	-0.694001776	-1.044143106	-0.754780102	-1.328303818	-1.581298794	-0.694001776	-1.044143106	-0.756131634	-1.328303818	-1.581298794	-0.694001776	-1.04642941
23	-0.694001776	-1.071886877	-0.806979731	-1.328303818	-1.581298794	-0.694001776	-1.07310825	-0.806979731	-1.328303818	-1.581298794	-0.694001776	-1.074661254
24	-0.694001776	-1.104708508	-1.088057656	-1.328303818	-1.581298794	-0.694001776	-1.096856271	-1.093909811	-1.328303818	-1.581298794	-0.694001776	-1.096856271

# 분석 방법

본 연구는 크게 비정상 신호 구간 탐색과 비정상 신호 구간 분류로 구성됨

## 비정상 신호 구간 탐색

- LSTM Auto Encoder & Control Chart
- Holt's Model & SOM

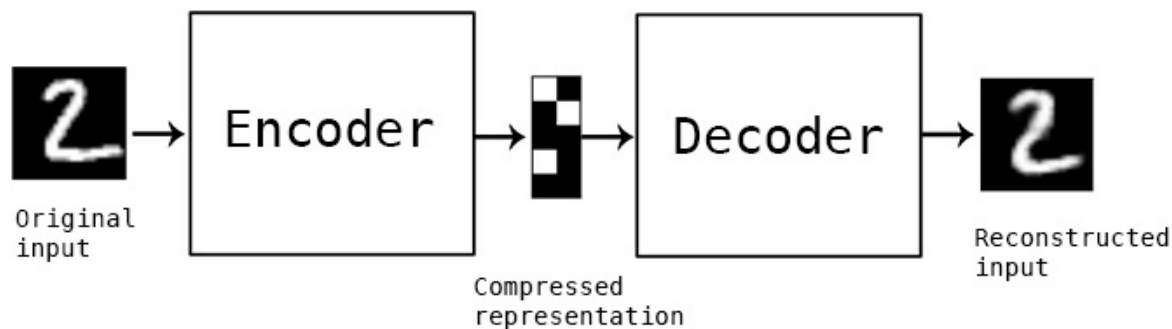
## 비정상 신호 구간 분류

- Convolutional Neural Network
- LSTM RNN
- Random Forest
- Gradient Boosting
- LSTM RNN + CNN

# 비정상 신호 구간 탐색 방법-1

비정상 신호 구간 탐색 첫번째 방법: **LSTM Auto Encoder**와 **Control Chart**를 사용

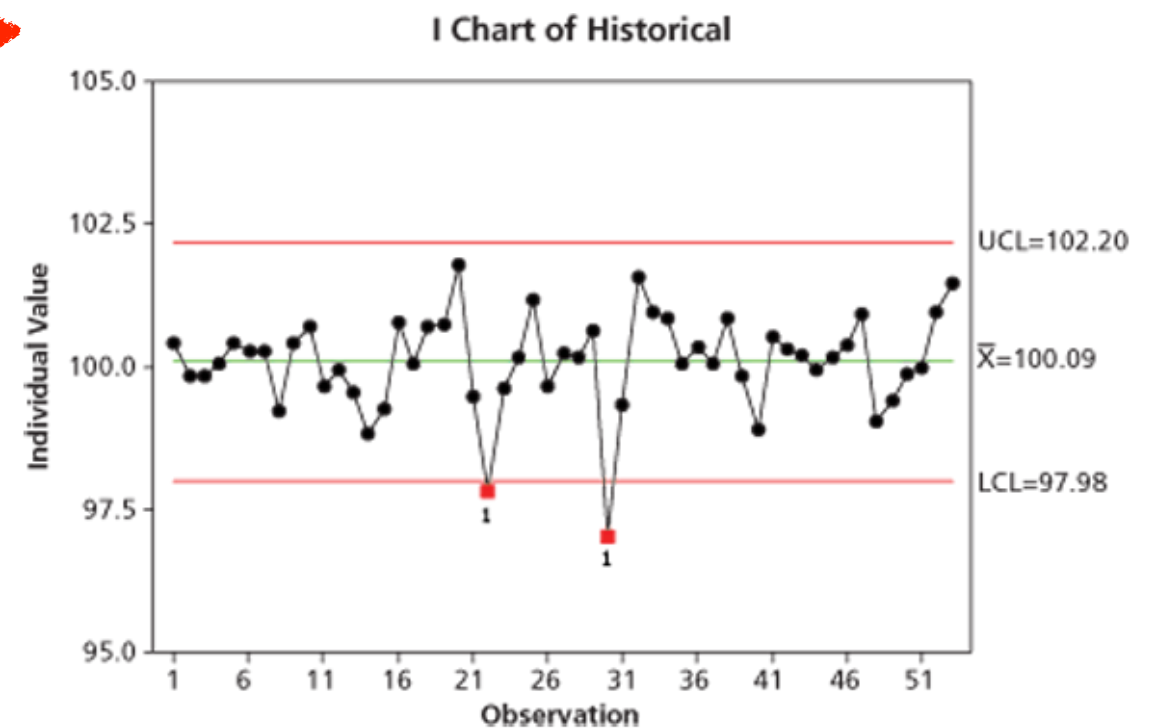
## LSTM Auto Encoder



- ▶ 데이터의 패턴을 찾아 다수의 패턴에 가장 유사하게 encoding
- ▶ Input이 encoding을 통해 hidden layer를 통과하며 각각 weight가 부과되어 output으로 decode 되어 나옴
- ▶ input과 output값의 차이가 크면 전체 패턴에서 많이 벗어난 값
- ▶ auto encoder에 LSTM layer를 추가해서 sequence를 반영해서 돌아갈 수 있게끔 함

- ▶ input에 value값 그대로 대입 (1500\*2325)  
\* <problem signal 1 기준>
- ▶ input 과 output의 difference, 즉 300프레임씩의 에러값들을 도출

## Control Chart



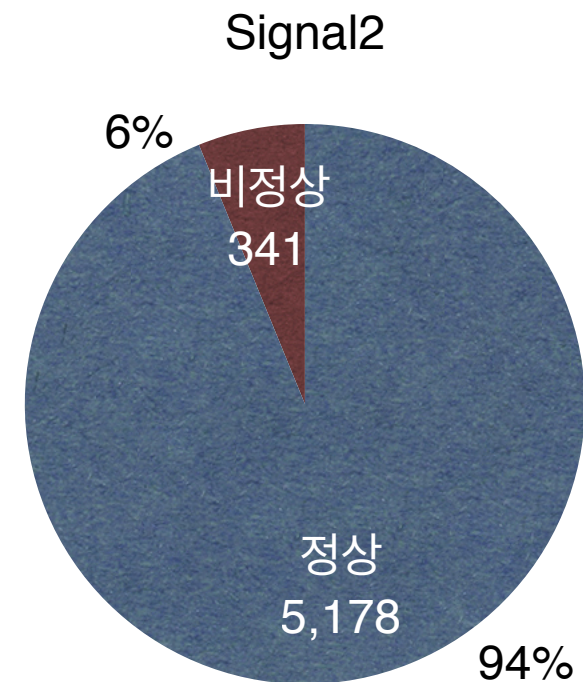
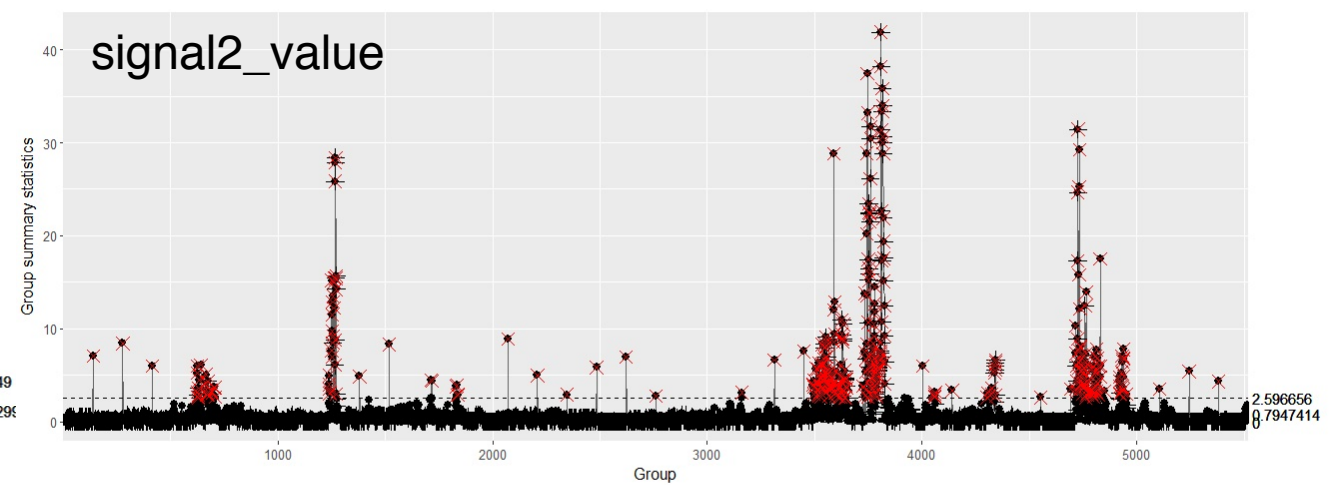
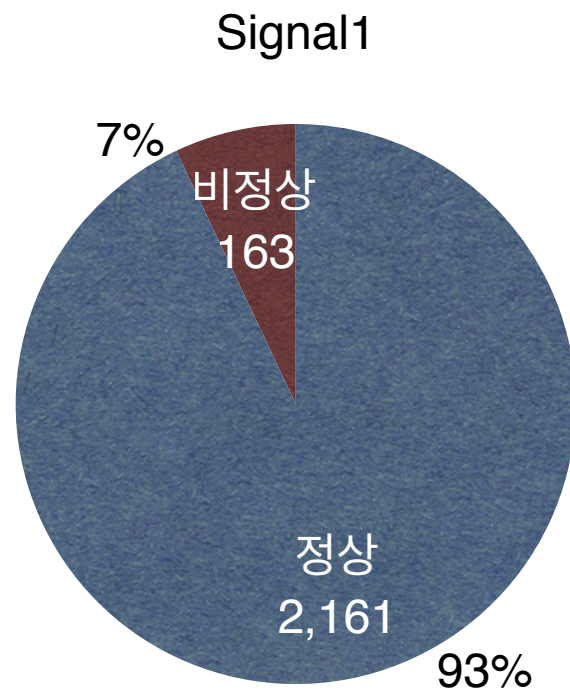
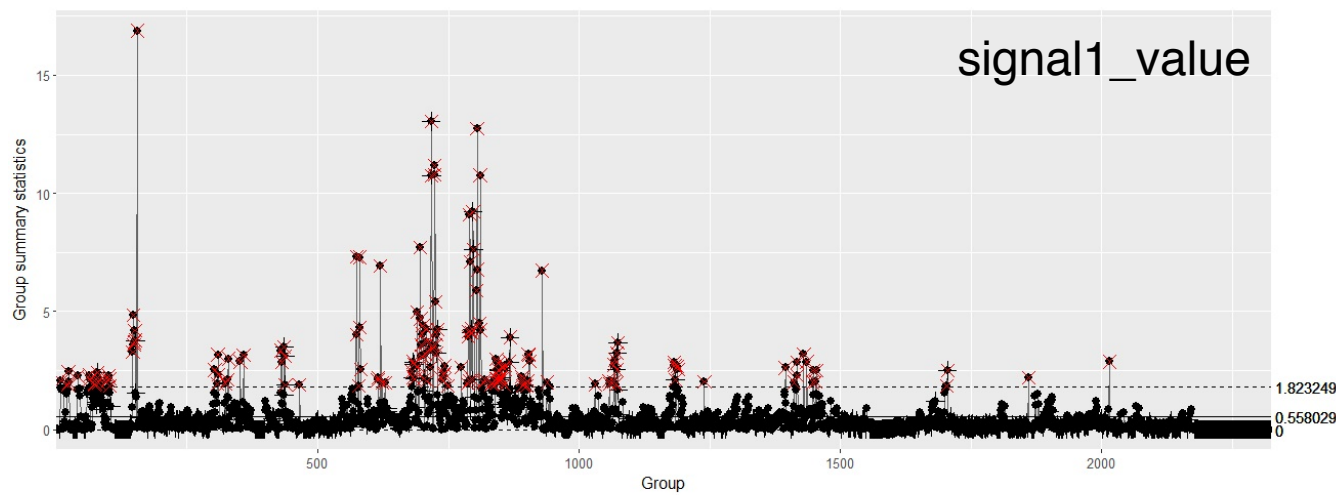
- ▶ I-MR Control Chart 이용 (Individual Moving Range)
- ▶ UCL: Upper Control Limit
- ▶ LCL: Lower Control Limit
- ▶ 개별 관측치의 평균  $\pm$  이동범위의 평균 으로 UCL, LCL결정
- ▶ UCL과 LCL을 벗어나는 측정값을 outlier라고 flag

**Auto encoder의 에러값들의 절댓값의 평균을  
control chart로 분석**



# 비정상 신호 구간 탐색 결과

LSTM Auto Encoder와 Control Chart사용한 비정상 신호 구간 탐색

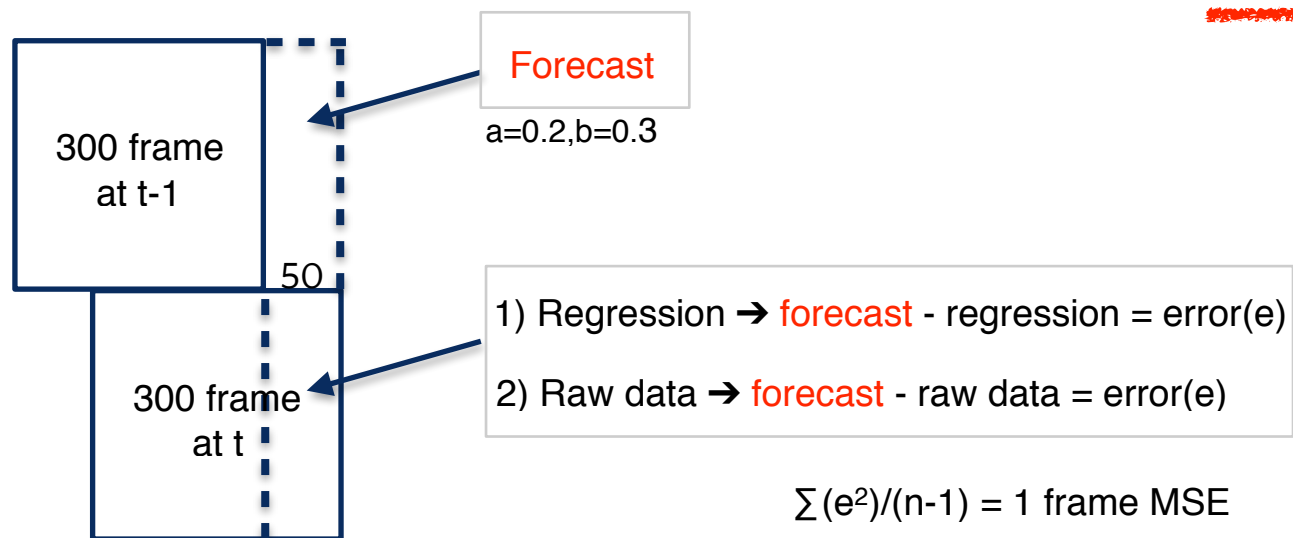




# 비정상 신호 구간 탐색 방법-2

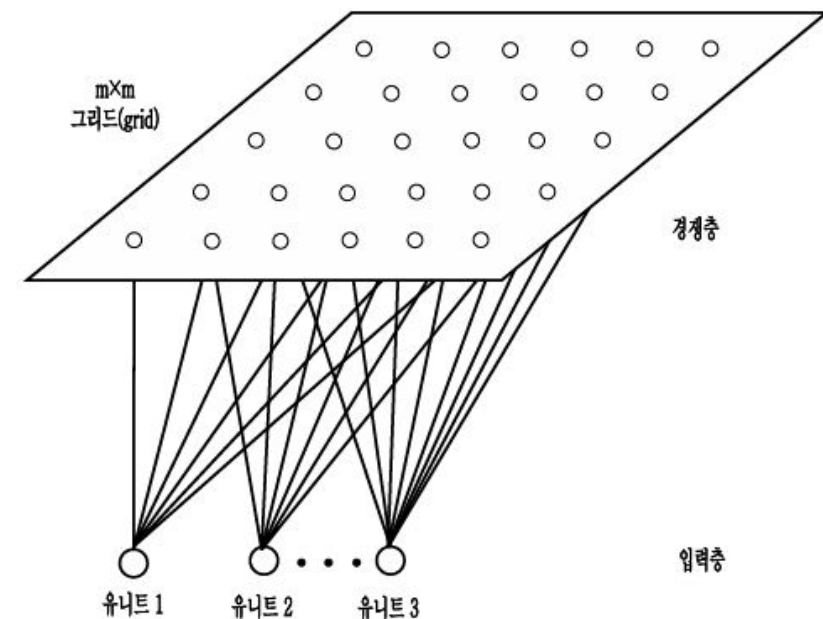
비정상 신호 구간 탐색 두번째 방법: **Holt's Model**과 **SOM**을 사용

## Forecasting: Holt's Model



- ▶ 데이터 값들의 변화 추이를 Trend로 이해하고, Holt's model을 적용하여, 앞선 프레임의 데이터를 바탕으로 다음 1초를 Forecast (이 때, 일반보정계수=0.2, 트렌드보정계수=0.3으로 임의 설정)
- ▶ Forecast한 데이터와 그 다음 프레임의 1초간의 데이터들 사이의 오차를 구하여 각 프레임마다 MSE (Mean Square Error) 값 도출  
비교1) t frame을 regression한 다음 forecast - regression  
비교2) forecast - raw data
- ▶ 변수들에 대해 개별적으로 구하여, 5\*2325 frame (data set 2는 5520 frame)으로 전처리함.

## Clustering: SOM(Self Organizing Map)



- ▶ Input layer와 competitive layer로 구성되어, input에서 competitive의 방향으로 완전 연결 되어 있음. Competitive 층 내에는 뉴런의 연결강도 vector가 임의 값을 가짐
- ▶ Competitive Learning을 통해, 연결강도 벡터와 입력 벡터가 얼마나 가까운지를 계산하여, 거리가 가장 가까운 뉴런이 승리해, 이 뉴런과 인접 뉴런만이 입력 벡터에 대해 학습하게 됨
- ▶ 승자 뉴런을 결정한 후에, 뉴런의 연결 강도를 정하여 이에 따라 이웃 반경을 줄여가면서 모든 뉴런이 반복 학습하게 됨.

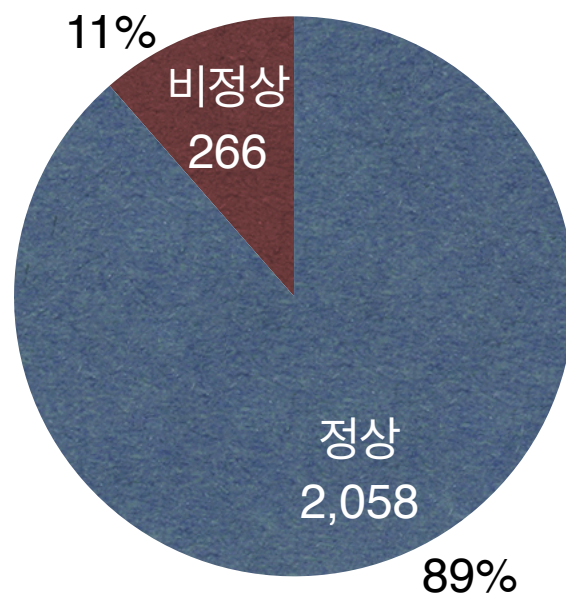
# 비정상 신호 구간 탐색 결과

## Holt's Model과 SOM사용한 비정상 신호 구간 탐색

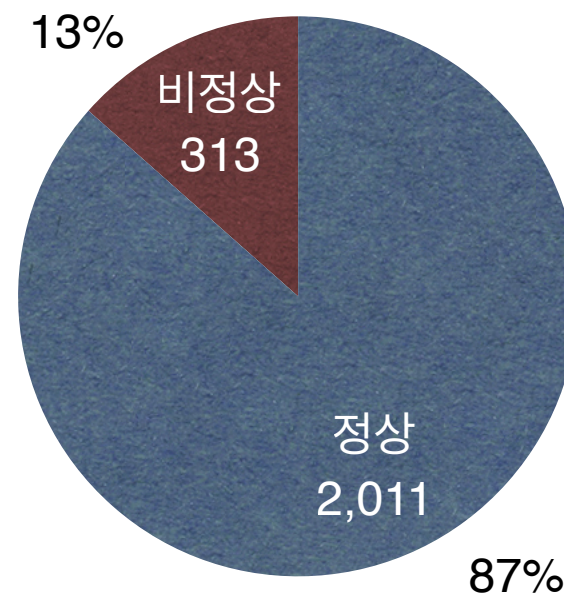
- ✓ cluster 개수: 36개
- ✓ 이유: 기어 6단 X 악셀페달 유지, 상승, 하락 X 정상, 비정상

- ✓ signal 1, 2모든 결과에서 한개의 cluster에 전체 데이터의 약 89%의 데이터가 모여있음
- ✓ 이 하나의 cluster가 정상, 나머지 소수의 cluster들이 비정상 신호일 것으로 예측
- ✓ 전체적으로 약 11%의 비정상과 89%의 정상 신호

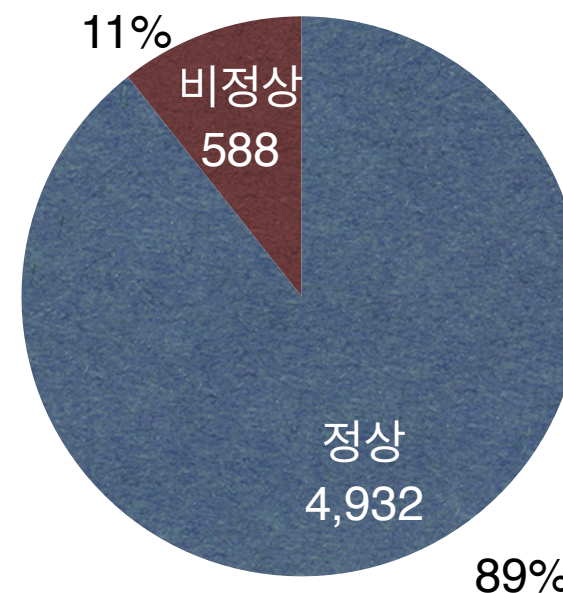
Signal1\_regression



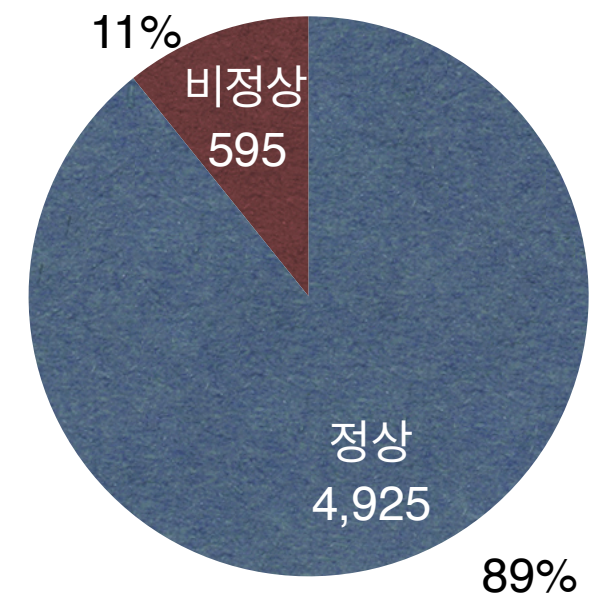
Signal1\_raw



Signal2\_regression



Signal2\_raw



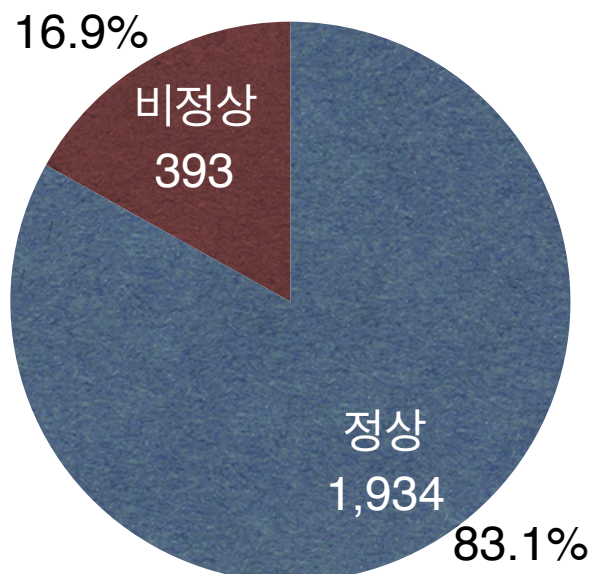
# 비정상 신호 구간 탐색 결과

## 비정상 신호 구간 탐색

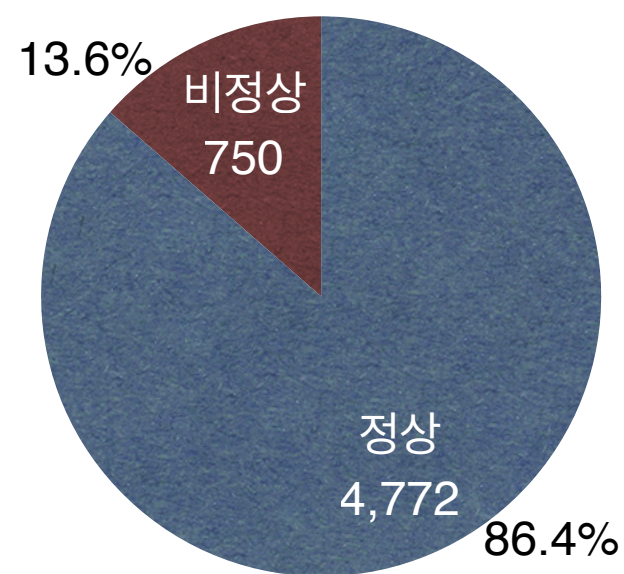
- ✓ LSTM Auto Encoder와 Control Chart를 사용해서 찾아낸 비정상 신호: 1번
- ✓ Holt's Model 과 SOM을 사용해서 찾아낸 비정상 신호: 2번

- ✓ 2번은 raw 와 regression의 교집합으로 선정
- ✓ 1번과 2번의 합집합을 최종 비정상 신호로 결정
- ✓ 이유: 한가지 기법 안에서는 교집합으로 함으로써 확실한 비정상 신호를 잡아낼 수 있고 다른 기법과 합칠 때에는 합집합으로 함으로써 한 기법이 놓친 비정상 신호까지 잡아낼 수 있다고 판단

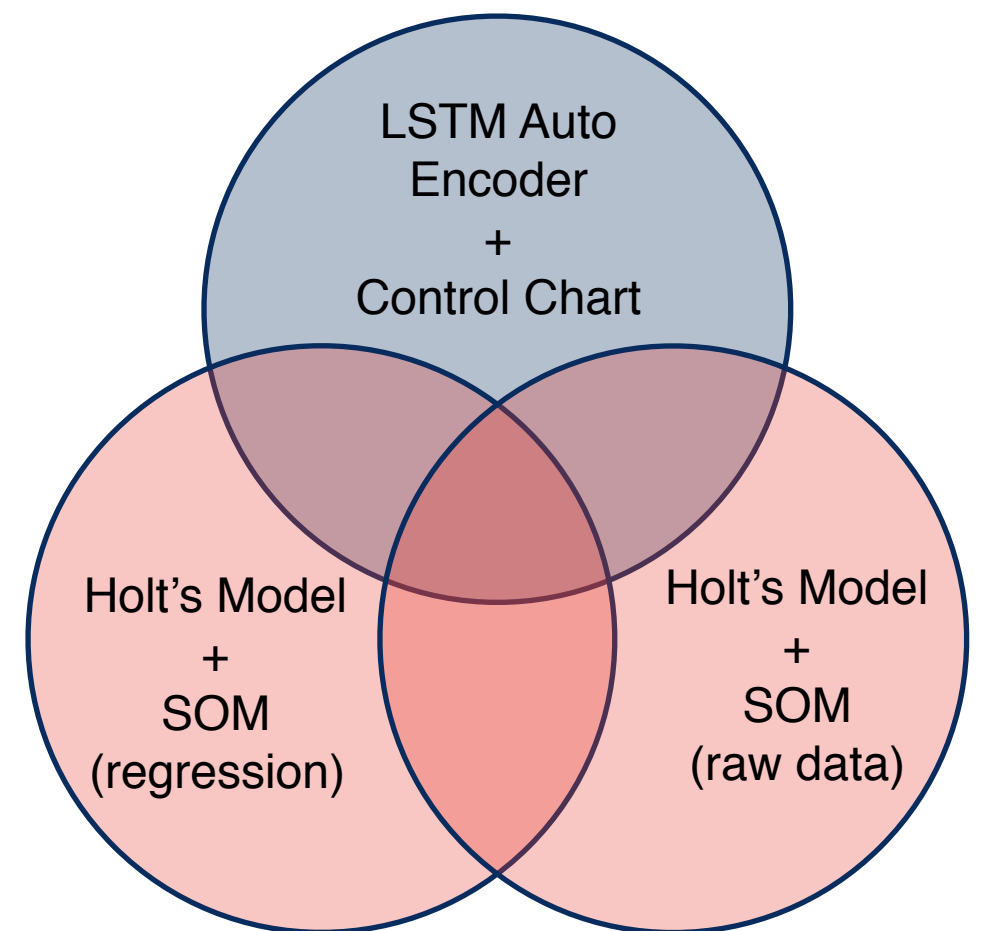
Signal1\_final



Signal2\_final



## Final





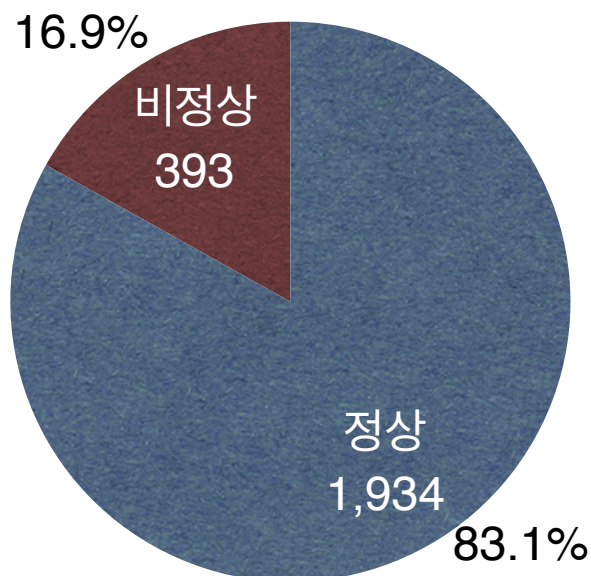
# 비정상 신호 구간 탐색 결과

## 비정상 신호 구간 탐색

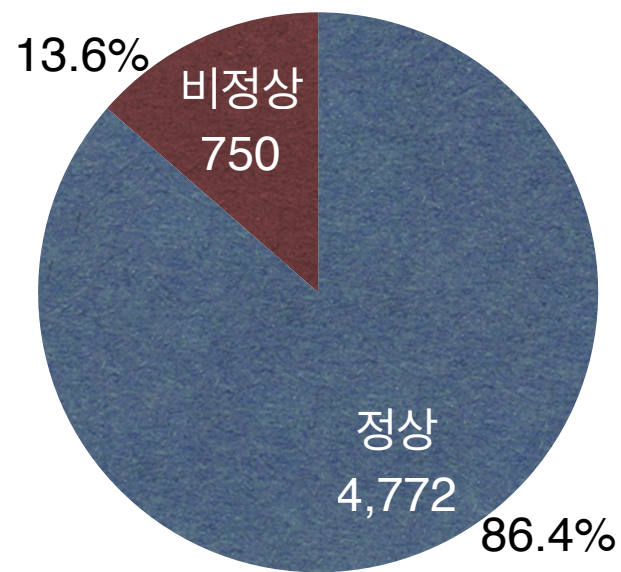
- ✓ LSTM Auto Encoder와 Control Chart를 사용해서 찾아낸 비정상 신호: 1번
- ✓ Holt's Model 과 SOM을 사용해서 찾아낸 비정상 신호: 2번

- ✓ 2번은 raw 와 regression의 교집합으로 선정
- ✓ 1번과 2번의 합집합을 최종 비정상 신호로 결정
- ✓ 이유: 한가지 기법 안에서는 교집합으로 함으로써 확실한 비정상 신호를 잡아낼 수 있고 다른 기법과 합칠 때에는 합집합으로 함으로써 한 기법이 놓친 비정상 신호까지 잡아낼 수 있다고 판단

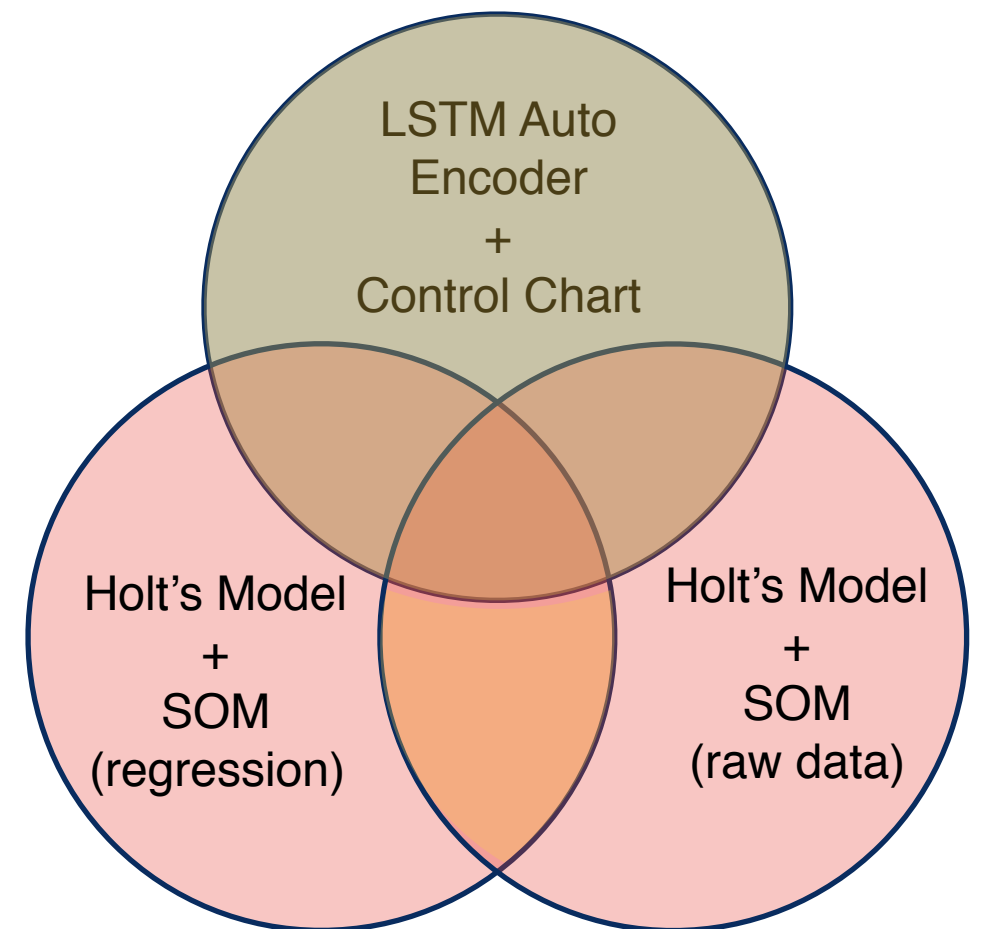
Signal1\_final



Signal2\_final

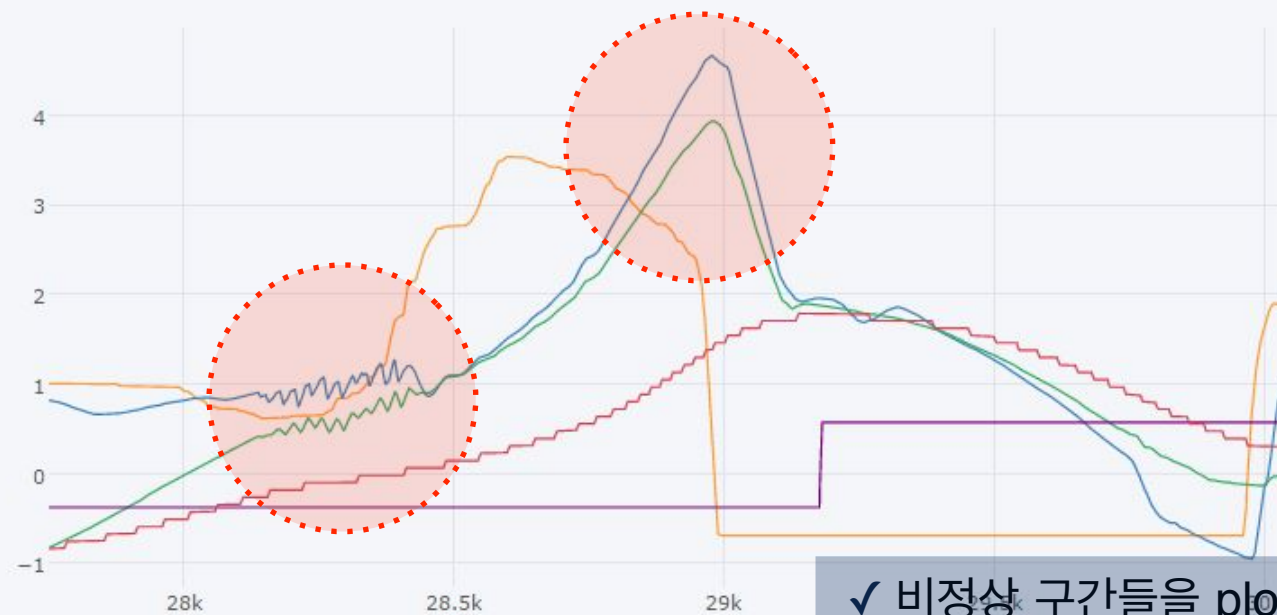


## Final

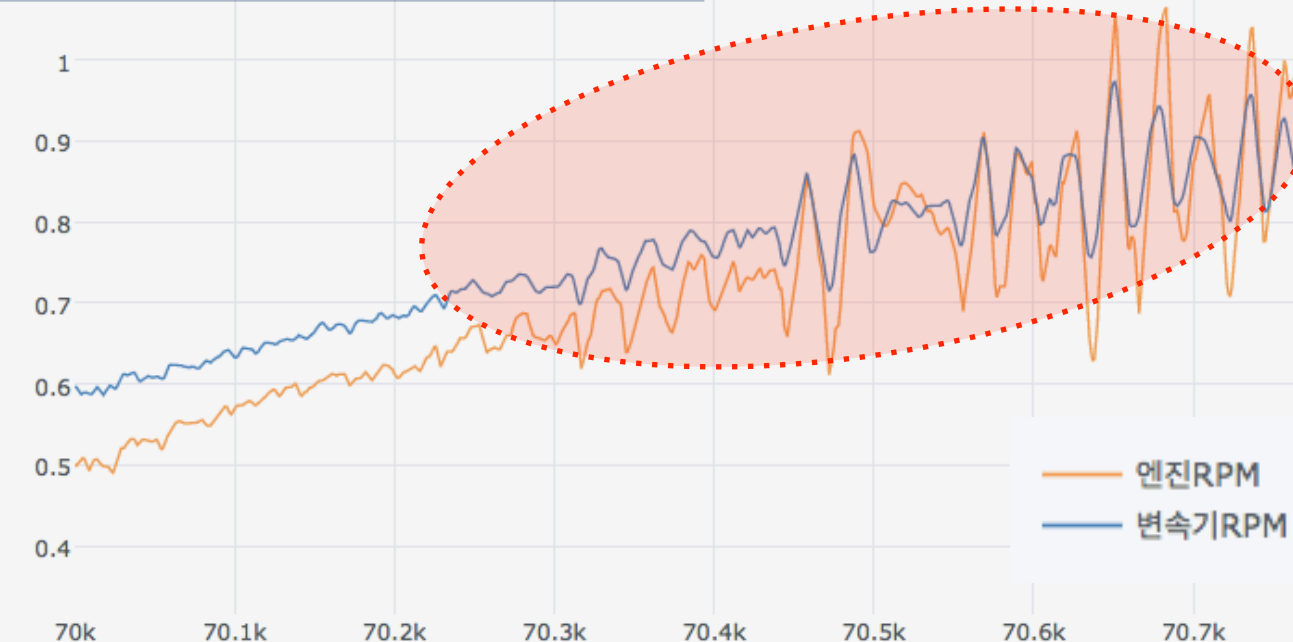


# 비정상 신호 구간 탐색 결과

## 비정상 신호의 검증



✓ 비정상 구간들을 plotting했을 때 엔진과 변속기 떨림 또는 튀는 현상을 잘 잡아낸 것을 볼 수 있음



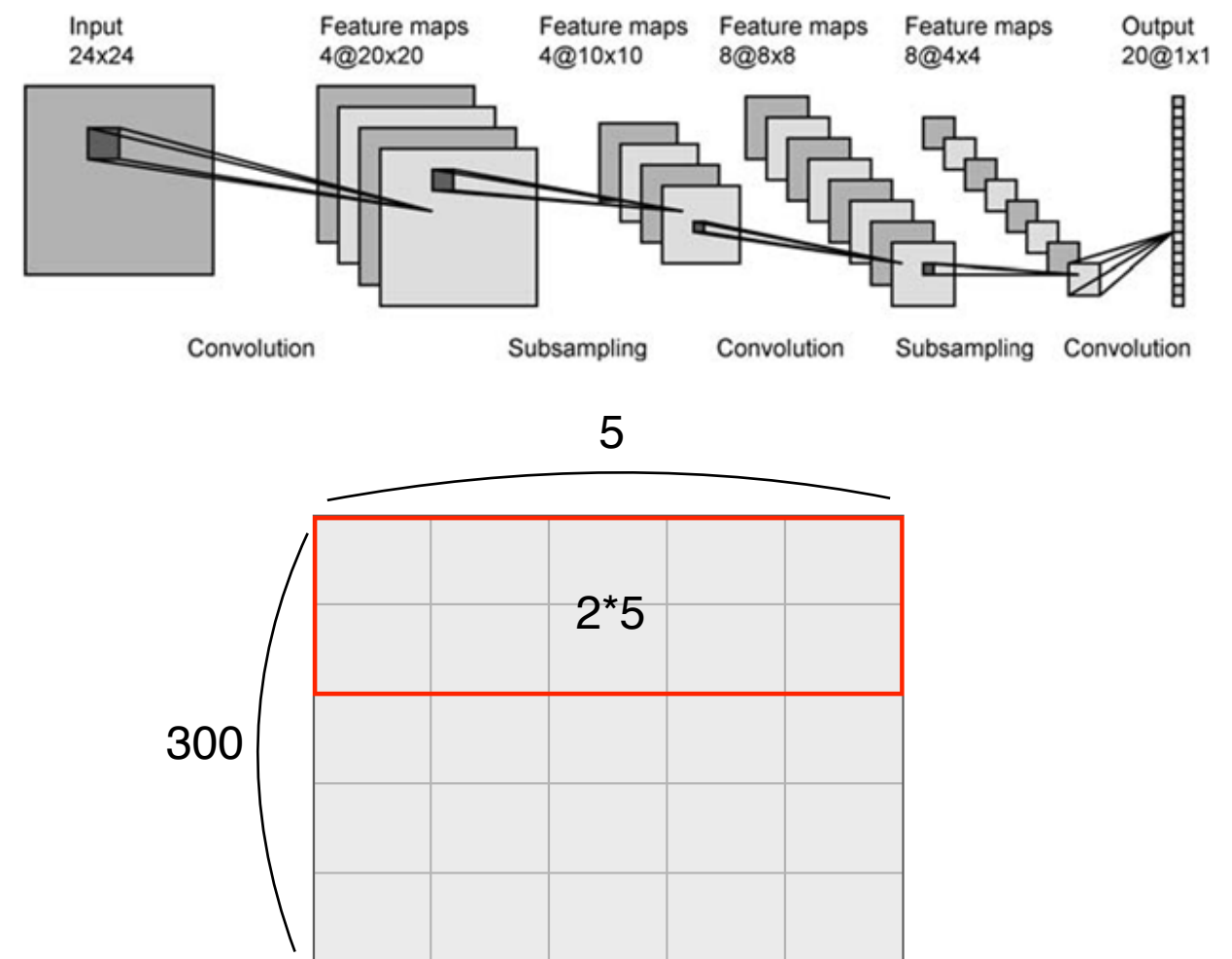
# 비정상 신호 구간 분류 방법

비정상 신호 구간 분류는 5가지 방법의 Supervised Learning으로 진행, 각각의 결과 비교

## 1. Convolutional Neural Network

- ▶ 딥러닝의 한 종류로 1차원의 time series data에 적용하기에 적합
- ▶ 인풋 데이터가 CNN 을 통해 연산되며 계산(학습)되어지며 목적하는 작업의 성공률이 높도록 주위 값들을 반영해 중앙의 값을 변형
- ▶ Convolutional layer
- ▶ 작은 조각의 필터로 데이터를 훑으며 cost값에 따라 레이어의 웨이트를 조정해주며 마지막에는 pooling layer를 거치는데 이는 조정된 weight들의 크기를 줄여주는 역할을 합니다. 이후에는 일반적인 neural network와 같은 방법으로 진행됩니다.

- ▶ Parameter: 2 convolution layer, 5\*2 filter\_size, max\_pooling(k=2)





# 비정상 신호 구간 분류 방법

비정상 신호 구간 분류는 5가지 방법의 Supervised Learning으로 진행, 각각의 결과 비교

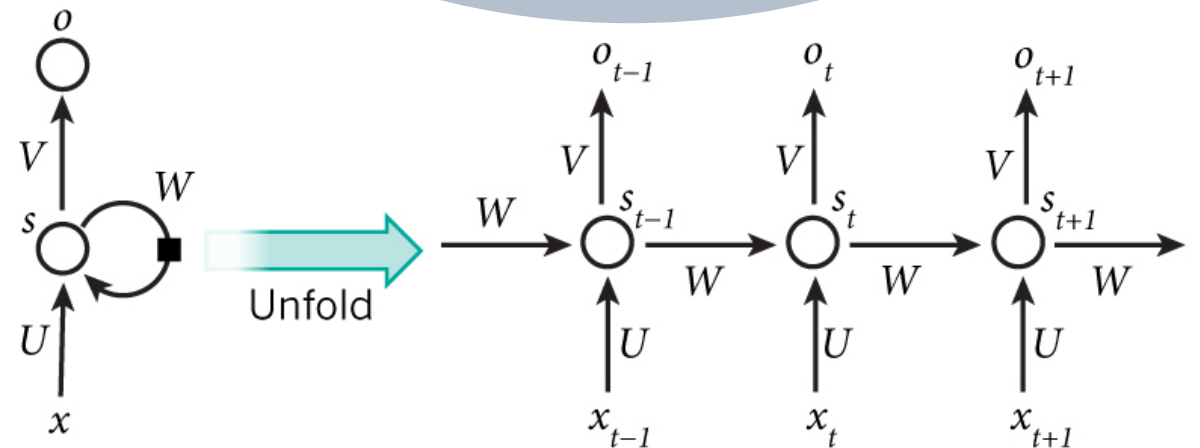
## 2. LSTM RNN (long short term memory Recurrent Neural Network)

- ▶ 기존 RNN의 약점을 보완하고자 만들어짐
- ▶ RNN은 backpropagation 해야 하는 state가 급격히 늘어나 vanishing gradient 라는 계속해서 미분해야 하는 값이 사라지는 문제가 생김 → 비교적 짧은 시퀀스만 효과적으로 처리할 수 있음
- ▶ LSTM은 backprop하는 과정에서 오차의 값이 더 잘 유지되게 도와줌 → 1000단계가 넘게 거슬러 올라갈 수 있음

- ▶ Parameter: 1 LSTM layer, 300 step\_size
- ▶ 한 번에 한 줄씩 보면서 예측

### RNN

- ▶ 순차적인 정보를 처리
- ▶ 기존의 신경망 구조에서는 모든 입력(과 출력)이 각각 독립적이라고 가정했지만 RNN은 동일한 태스크를 한 시퀀스의 모든 요소마다 적용하고, 출력 결과는 이전의 계산 결과에 영향을 받음
- ▶ RNN은 계산된 결과에 대한 "메모리" 정보를 갖고 있다고 볼 수도 있음



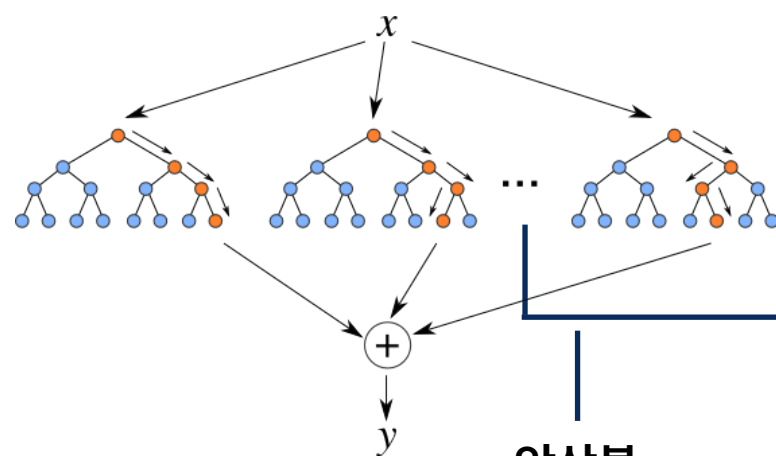
# 비정상 신호 구간 분류 방법

비정상 신호 구간 분류는 5가지 방법의 Supervised Learning으로 진행, 각각의 결과 비교

## 3. Random Forest

- ▶ 여러 개의 결정 트리들을 임의적으로 학습하는 방식의 앙상블 방법
- ▶ 핵심적인 특징: 임의성(randomness)에 의해 서로 조금씩 다른 특성을 갖는 트리들로 구성
- ▶ 임의화는 각 트리들의 훈련 과정에서 진행
- ▶ 배깅과 임의 노드 최적화 → 서로 동시에 사용되어 임의화 특성을 더욱 증진 시킬 수 있다.

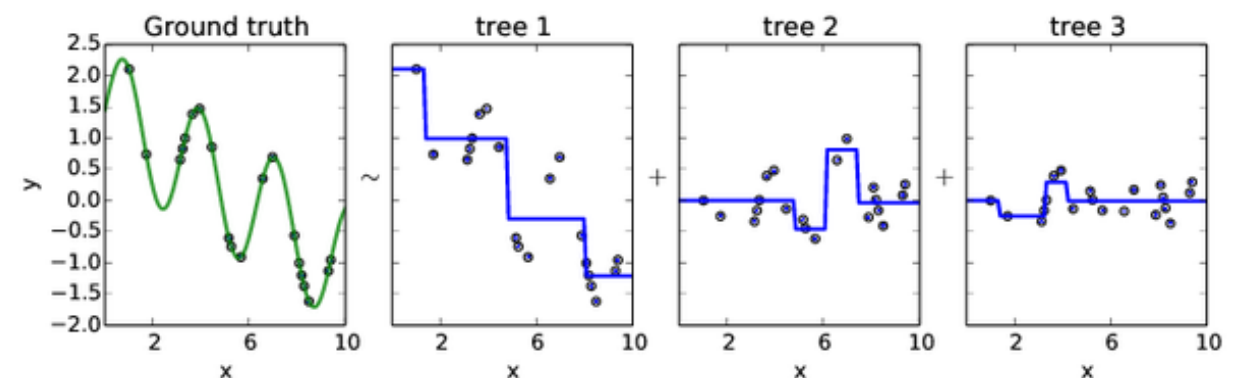
- ▶ parameter: max\_feature: 38, forest 하나당 tree: 10개



## 4. Gradient Boosting

- ▶ 최적화 알고리즘의 하나인 기울기 강하(gradient descent) 알고리즘을 통하여 지수 손실함수 이외의 다양한 손실함수에서 부스팅 알고리즘을 사용
- ▶ Boosting은 이전 분류기의 학습 결과를 토대로 다음 분류기의 학습 데이터의 샘플가중치를 조정해 학습을 진행하는 방법

- ▶ parameter: xgboost의 default



- ✓ Bagging의 경우 각각의 분류기들이 학습시에 상호 영향을 주지않고 학습이 끝난 다음 그 결과를 종합하는 기법
- ✓ Boosting은 이전 분류기의 학습 결과를 토대로 다음 분류기의 학습 데이터의 샘플가중치를 조정해 학습을 진행하는 방법

# 비정상 신호 구간 분류 결과

5가지의 비정상 신호 구간 분류 알고리즘을 이용한 결과를 비교

- ✓ 75%의 데이터로 학습시키고 25%의 데이터로 분류 실시
- ✓ 총 7847 개 데이터 중  
train data : 5885 / test\_data: 1962
- ✓ 100번 testing 평균
- ✓ test마다 resampling

	CNN	LSTM RNN	Random Forest	Gradient Boosting	LSTM RNN + CNN
정확도	0.952399	0.958079	0.953542	0.952982	0.949479



# 비정상 신호 구간 분류 결과

5가지의 비정상 신호 구간 분류 알고리즘을 이용한 결과를 비교

- ✓ 75%의 데이터로 학습시키고 25%의 데이터로 분류 실시
- ✓ 총 7847 개 데이터 중  
train data : 5885 / test\_data: 1962
- ✓ 100번 testing 평균
- ✓ test마다 resampling

	CNN	LSTM RNN	Random Forest	Gradient Boosting	LSTM RNN + CNN
정확도	0.952399	0.958079	0.953542	0.952982	0.949479

LSTM RNN: 그 전 값을 고려해서 가중치 갱신하기 때문에  
시계열 데이터를 처리하는데 가장 적합  
이 센서 데이터는 CNN처럼 한번에 뭉쳐서 보기 보다는 데이터 그대로의 시퀀스를 읽어주는게 해석하기에 좋음  
(앞뒤 상황에 영향을 많이 받음)

# 비정상 신호 구간 분류 결과

5가지의 비정상 신호 구간 분류 알고리즘을 이용한 결과를 비교

- ✓ 75%의 데이터로 학습시키고 25%의 데이터로 분류 실시
- ✓ 총 7847 개 데이터 중  
train data : 5885 / test\_data: 1962
- ✓ 100번 testing 평균
- ✓ test마다 resampling

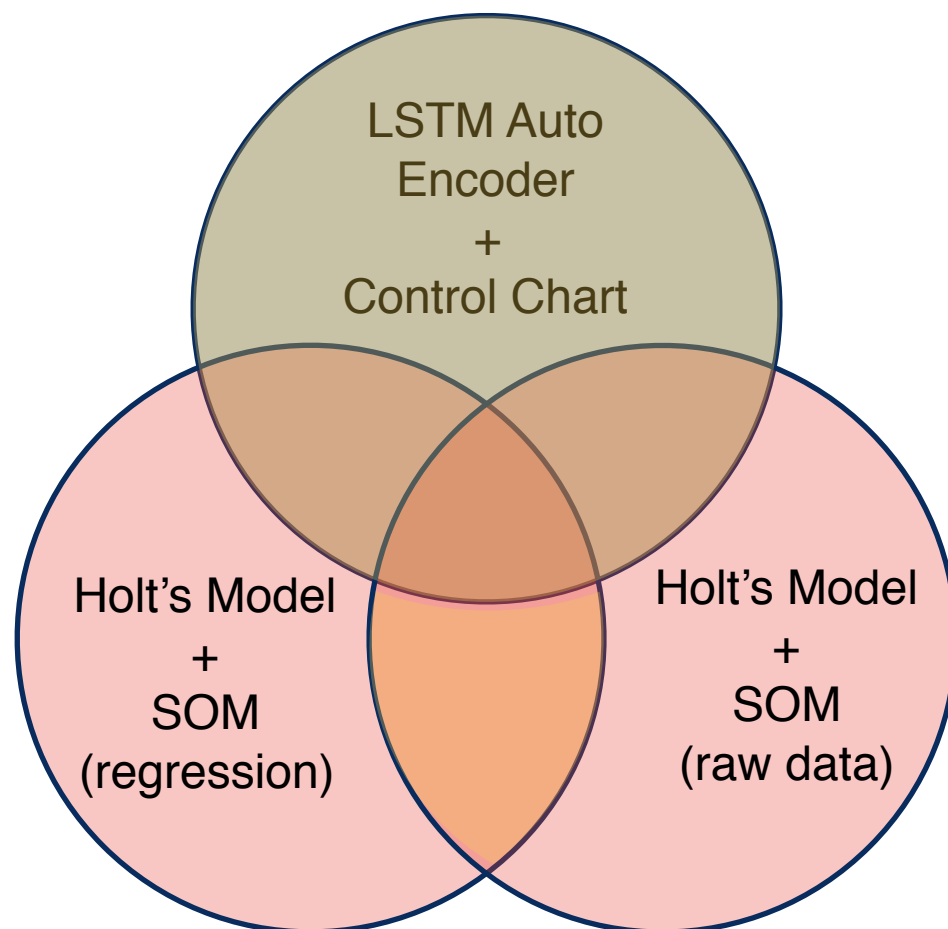
	CNN	LSTM RNN	Random Forest	Gradient Boosting	LSTM RNN + CNN
정확도	0.952399	0.958079	0.953542	0.952982	0.949479

LSTM RNN + CNN으로 했을 시 LSTM RNN보다 정확도가 떨어졌지만 모델이 복잡해졌기 때문에 over fitting 이 되었기 때문이고 데이터의 수가 많아져서 over fitting을 극복한다면 LSTM RNN + CNN의 정확도가 더 높게 나올 것으로 예상

# 요약

## 비정상 신호 구간 탐색

Final



## 비정상 신호 구간 분류

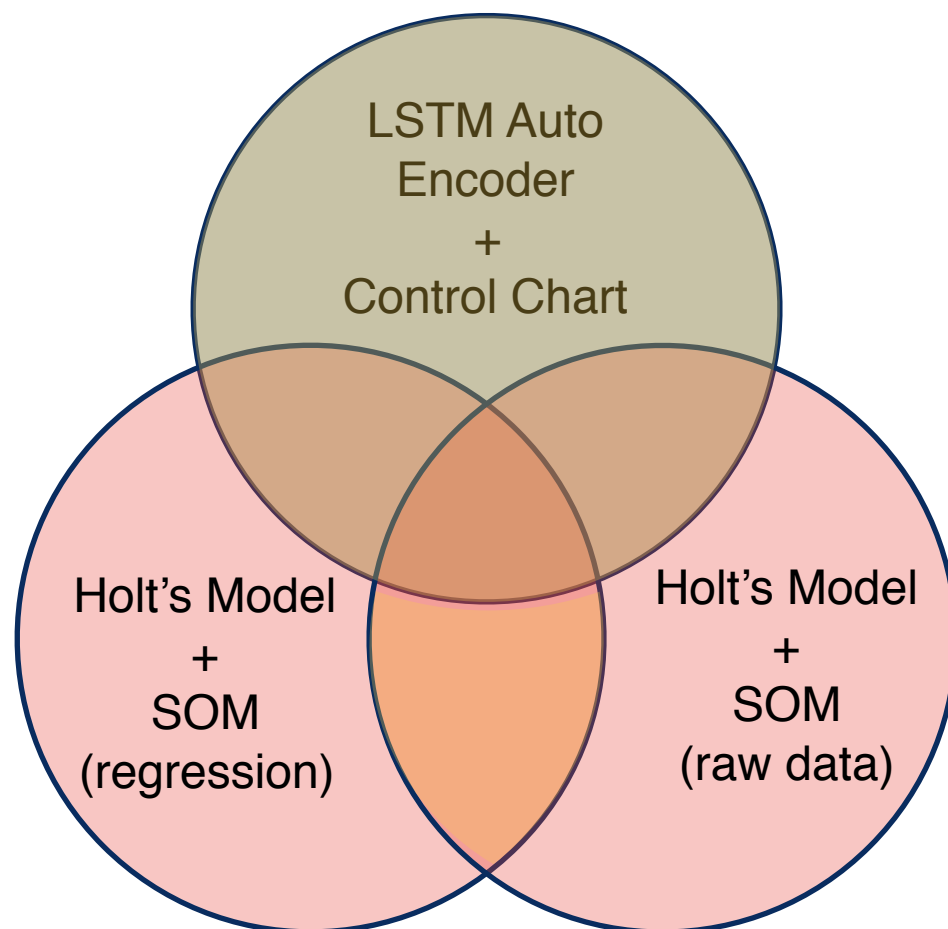
	CNN	LSTM RNN	Random Forest	Gradient Boosting	LSTM RNN + CNN
정확도	0.952399	0.958079	0.953542	0.952982	0.949479



# 요약

## 비정상 신호 구간 탐색

Final



## 비정상 신호 구간 분류

	CNN	LSTM RNN	Random Forest	Gradient Boosting	LSTM RNN + CNN
정확도	0.952399	0.958079	0.953542	0.952982	0.949479

# 기대효과 및 한계

## 기대효과

현실에서는 정상/비정상이 정의되어 있지 않은 경우가 많다. 본 연구를 통해서 비정상 구간 분류 알고리즘을 개발하여 정상/비정상이 정의하기 어려운 상황에서도 적절하게 기계가 비정상적으로 작동하는 것을 감지할 수 있는 것이 가능할 것이라 기대된다. 더 나아가 이에 대한 원인을 찾는 역학조사로 바로 이어질 수 있게끔 하는 것이 가능해질 것이라고 생각한다.

## 한계

1. case별 이상 현상을 세부적으로 클러스터링 하지 못함
2. 이상을 나타내는 원인 변수에 대한 탐지가 미흡
3. 비정상 신호에 대한 검증 방법 부족
4. 데이터 수가 적어 LSTM RNN + CNN모델에서 overfitting이 일어남

# 참고 문헌

- [1] 주태우, 김성범 (2013). 다중 정상 하에서 단일 클래스 분류기법을 이용한 이상치 탐지. 대한산업공학회지, 39(2), 82-89.
- [2] 남윤진, 신동일, 신동규 (2015). 센서 데이터 분류에 대한 분류 알고리즘 성능 비교. 한국정보과학회 학술발표논문집, 1891-1893.
- [3] 백수정, 김덕영(2013). 자동차 엔진 고장 감지 기법 비교 연구. 한국 CDE학회 학술발표회 논문집, 243-248.
- [4] 전제성, 구자갑, 박창목(2015). 규칙기반 및 상관 분석 방법을 이용한 시계열 계측 데이터의 이상치 판정. 한국지반환경공학회 논문집, 16(5), 43-53. 1891-1893.
- [5] A. Krizhevsky, I. Sutskever, and G. E. Hinton (2012). ImageNet Classification with Deep Convolutional Neural Networks, NIPS
- [6] S. Hochreiter, and J. Schmidhuber (1997). Long short-term memory. Neural Comput., 9(8):1735–1780,.
- [7] N. Srivastava, E. Mansimov, and R. Salakhutdinov (2016). Unsupervised Learning of Video Representations using LSTMs
- [8] L. Breiman (2001). Random forests. Machine Learning, 45:5–32,.

## 자료 출처

- [1] 8p auto encoder 사진 1- <https://blog.keras.io/building-autoencoders-in-keras.html>
- [2] 10p neural network 사진- <http://www.opennn.net>
- [3] 10p convolutional neural network 사진- <https://stats.stackexchange.com/questions/180850/how-are-filters-and-activation-maps-connected-in-convolutional-neural-networks>
- [4] 12p gradient boosting 사진 - <https://www.quora.com/How-would-you-explain-gradient-boosting-machine-10p-learning-technique-in-no-more-than-300-words-to-non-science-major-college-students>
- [5] 12p random forest 사진-<https://kgpdag.wordpress.com/>
- [6] 9p “Kohonen Network”, [http://www.aistudy.com/neural/som\\_kim.htm](http://www.aistudy.com/neural/som_kim.htm) (2017.05.10)

한국 신뢰성 학회-현대자동차 퀴즈 세션

감사합니다

