DACON

진동데이터 활용 충돌체 탐지 AI 경진대회

팀 : 두부

목차

디렉터리 구조

Data set

‘./DACON_KAERI/sample_submission’
‘./DACON_KAERI/test_features’
‘./DACON_KAERI/train_features’
‘./DACON_KAERI/train_target'

실행순서

1. DACON_KAERI_코드정리용(Lasso + RF)
2. DACON_KAERI_코드정리용(CNN)

Code

./DACON_KAERI_코드정리용(Lasso + RF)
./DACON_KAERI_코드정리용(CNN)

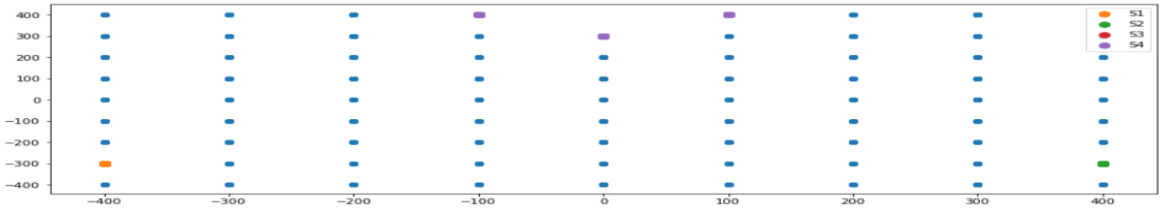
output

최종 submit file : Submission_final.csv

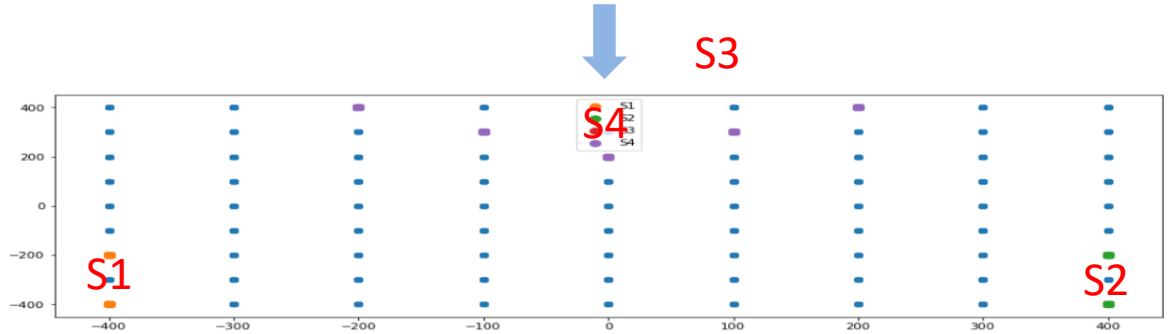
1. 좌표

좌표의 경우는 각 센서에 파장이 도착한 시간에 가장 큰 영향을 받음

--> 제일 먼저 도착한 파장이 도착한 센서와 그 시간을 통해 센서의 대략적 위치 유추 가능



각 센서에 제일 먼저 도착했으면서 0.00004초에 도달한 경우

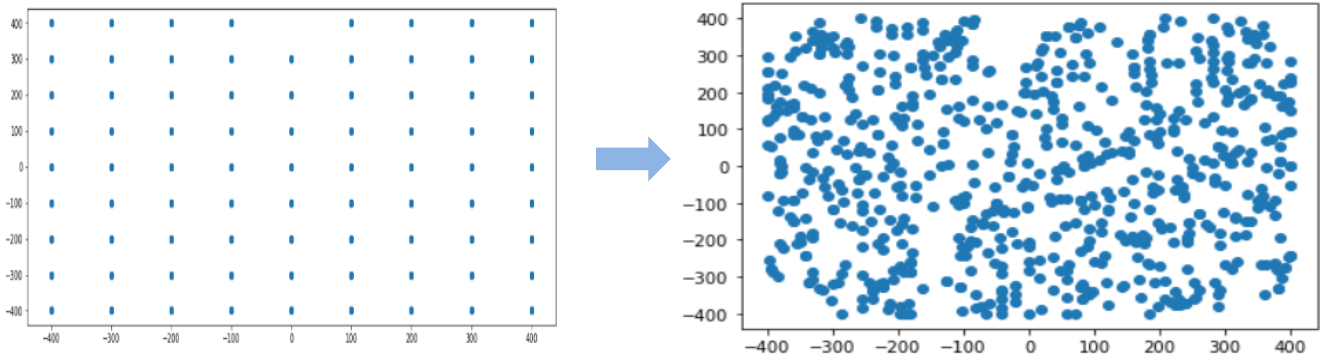


각 센서에 제일 먼저 도착했으면서 0.00008초에 도달한 경우

1. 좌표

하지만, 각 센서에 도착한 시간만으론 좌표를 맞추는데 한계가 존재

→ 센서에 도착한 시간이 비슷하더라도, 좌표값은 크게 다를 수 있음 (외삽의 문제)
→ 좌표 유추에도 센서에 도달한 파장의 정보가 필요



2.파장

파장의 모양에 영향을 주는 요소에는 크게 3가지 요소가 존재

1.파장이 센서에 도착한 시간

--> 센서에 파장이 빨리 도착할 수록 파장의 높낮이가 큼

2.V

--> V가 클수록 파장의 기울기가 가파름

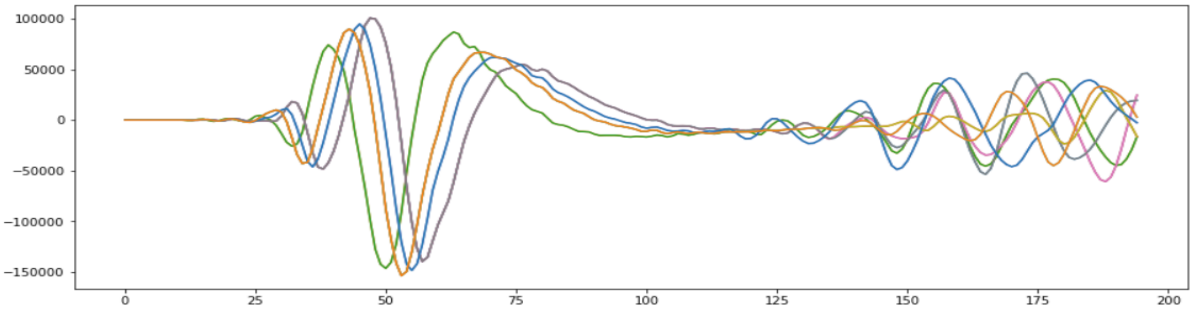
3.M

--> M과 파장의 진폭, 파장의 높낮이는 깊은 관련

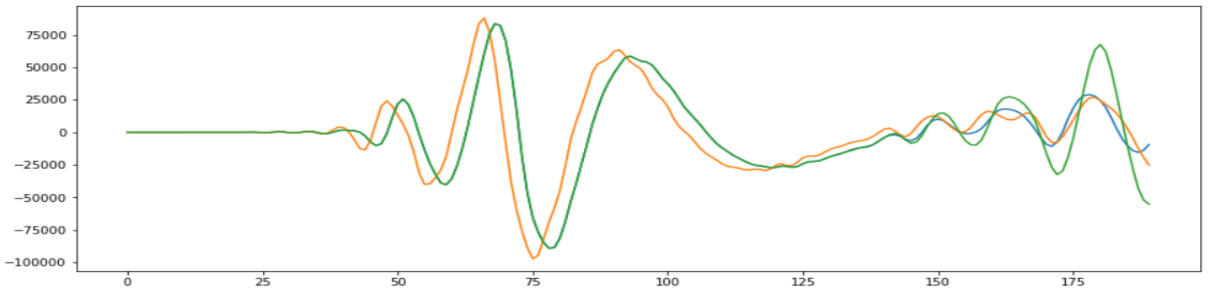
→ 이미지 데이터와 유사하게 풀이가 가능

2.파장

1. 파장이 도착한 시간에 따른 모양 (M = 0.2, V = 0.2 고정)



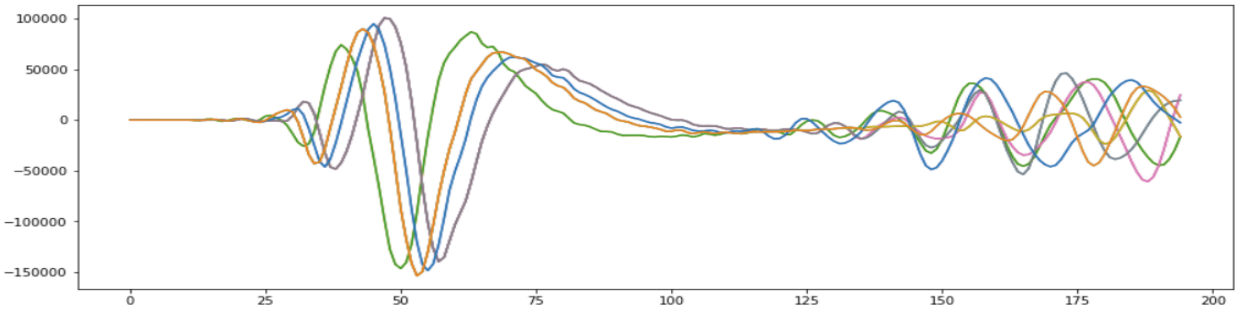
Time = 5



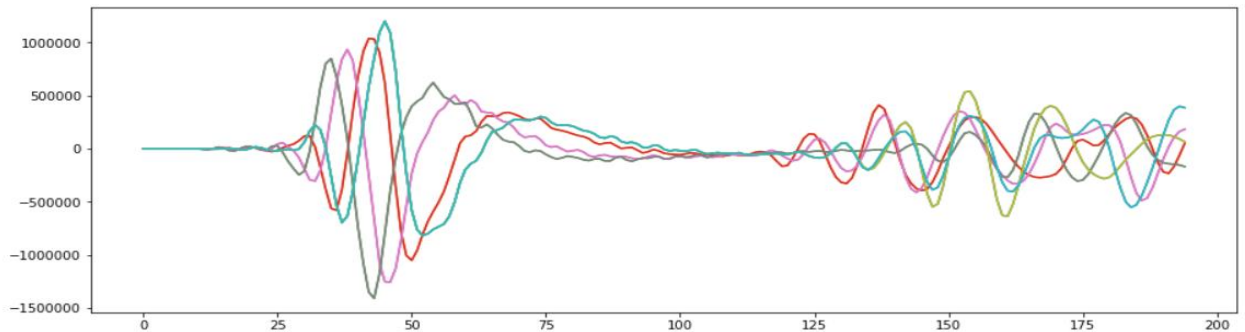
Time = 10

2.파장

1. V에 따른 모양 (Time=5, M=0.2)



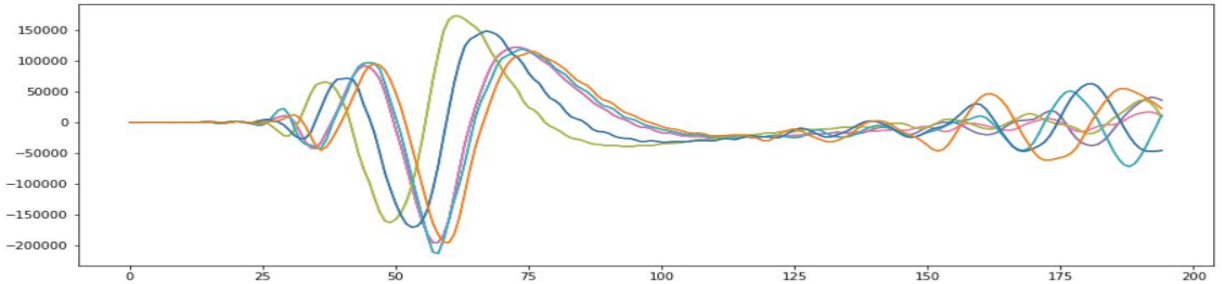
V = 0.2



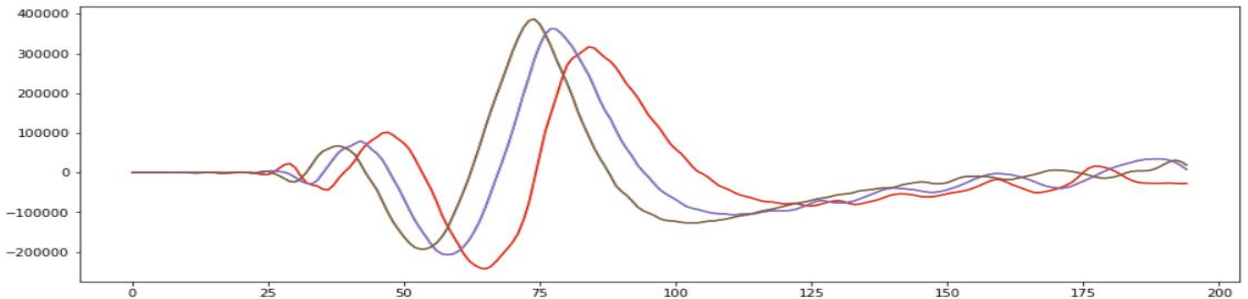
V = 1

2.파장

1. M에 따른 파장의 모양 -> 파장의 높낮이와 진폭 -> 적분값과 관련 (Time=5, V=0.2)



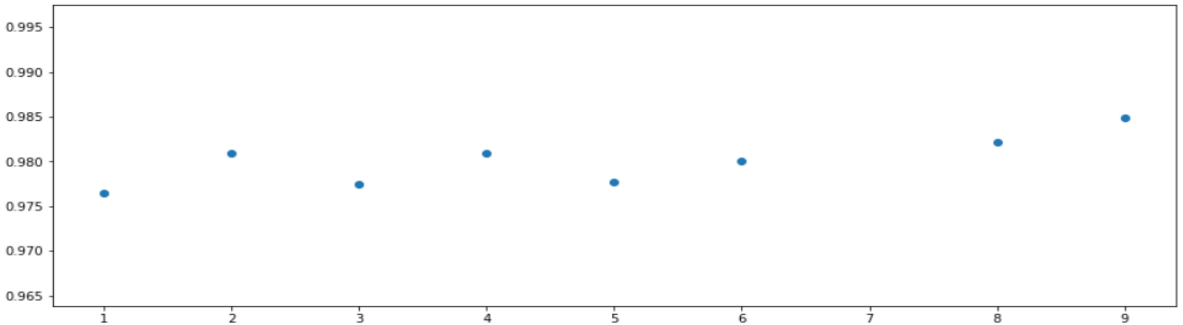
M = 25



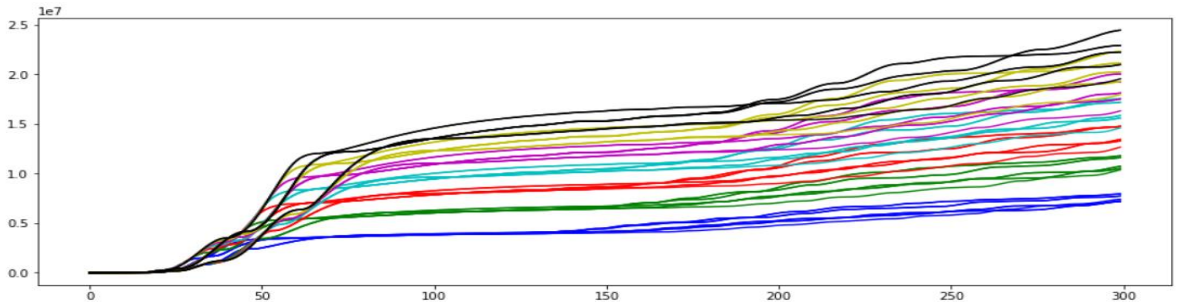
M = 175

2.파장

1. 센서가 도착한 시간별 파장의 기울기 max-min과 V의 상관관계



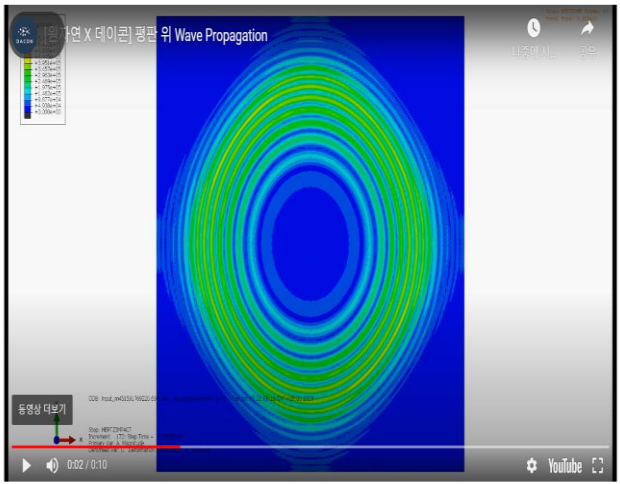
2.M에 따른 파장의 적분값 (처음 도달한 센서값의 absolute cumsum) (Time, V고정)



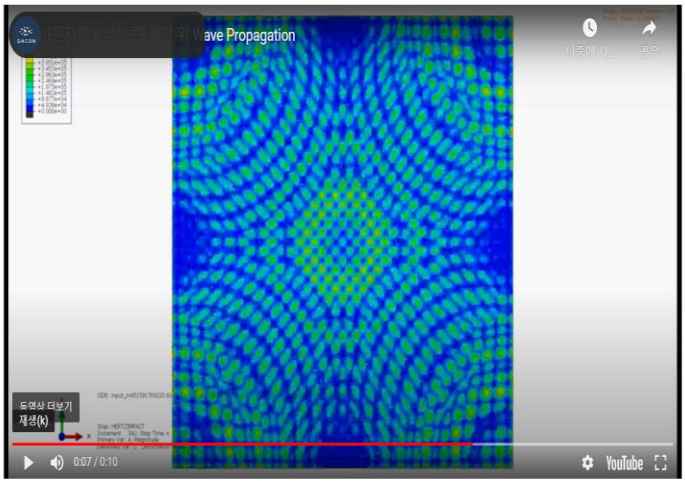
2.파장

센서에 파장이 닿은 후, 특정 시간이후의 파장은 반사파

→ 특정 시간 이후의 파장은 무의미한 정보를 가지고 있을 수 있음
→ 실험 결과 Time 200이하로 쓰는 것이 적절



반사파X



반사파O

3.결론

1.파장의 모양, 도착시간은 target 값과 깊은 관련이 있음

-> CNN과 같은 NN모형으로 접근 가능

2. 파장에서 추출한 통계량을 통해서도 접근 가능

-> Lasso, RF 와 같은 단순한 모형으로 접근 시도

3.특정 시간대 이후의 파장은 반사파로 추정

-> 특정 시간대 이후의 센서값은 모델링에 불필요 할 수도 있음

1.Lasso + RF

-Lasso, RF은 파장에서 추출한 통계량만을 가지고 학습

-정확한 통계량을 추출하기 위해, 각 sample별로 **최초 도착한 센서값만** 사용

-센서의 도착시간에 따라 통계량에 유의미한 차이가 있기 때문에 **도착 시간별로 모델링**

-Lasso + RF 앙상블

2. 모델링

1.Lasso + RF

1.V

Sample row data, Sample 별로 정규와 이후
-기울기 max – min-
-기울기 분산
-기울기 절대값 max
-기울기 절대값 평균
- 각 변수의 제곱, 세제곱 등등…

2.M

Sample row data, Sample 별로 정규와 이후
-기울기 max – min-
-기울기 분산
-기울기 절대값 max
-기울기 절대값 평균
-각 변수의 제곱, 세제곱 등등…
-특정 시간대까지의 적분값/V 예측값

3.X,Y

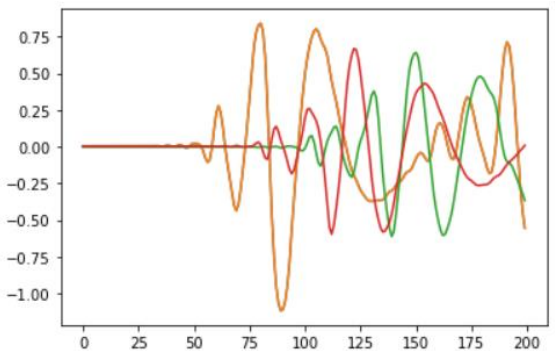
-각 센서별 도착한 시간
-각 센서별 도착 시간의, 각각의 차
-각 센서별 도착 시간의, 각각의 차 절대값
-각 센서별 도착 시간의 , 각각의 합

2.CNN

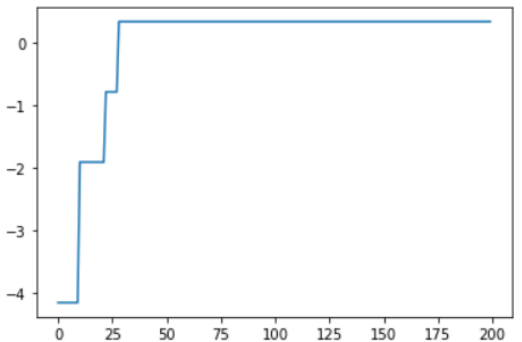
-row데이터 정규화 이후 학습

-각 시간마다 4개의 센서 중, **몇 개의 센서에 파장이 닿았냐를 알려주는 변수 생성**
(ex, 0…0001….11222….3333…444444) -> 센서가 닿은 시간을 직접으로 알려주기위함

-**특정시간 이후의 센서값은 사용하지 않음**
(ex 0~375개의 Time중, 0~200만 사용)



센서 1,2,3,4



파생변수

2.CNN

Case1 : CNN + 변수추가 + 5Fold + Cosine Scheduler + Time 200까지만

Case2 : CNN + 변수추가 + 5Fold + Time 200까지

Case3 : CNN + 5Fold + 변수추가 + regularizer

Case4 : CNN + 5Fold + Cosine Scheduler (base model)

3.앙상블

최종적으로
(Case1 + Case2)*0.5 + (Case3 + Case4 + LassoRF)*0.5 를 사용

3. 결과

1.Lasso + RF

LB : 0.018X

-> 최초 도달한 센서값만 사용했기에 한계점 존재
-> 세밀한 통계량을 추출하는 데에 있어 한계 존재

2.CNN

Case1 + Case2 LB : 0.0035
Case3 + Case4 + LassoRF LB : 0.0067

변수추가 + Time 200이하 사용을 통해 단일 모델로도 0.0035 성능

3.앙상블

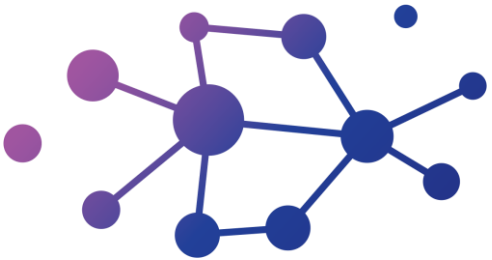
LB : 0.0025

(CNN Case1 + CNN Case2)*0.5 + (CNN case3 + CNN case4 + LassoRF)*0.5

결론

1.EDA를 통해 적절한 변수와 적절한 데이터 사용만으로도 좋은 성능을 거둘 수 있었음

2.다양한 시도를 해보았지만, M,V의 범위를 각각 [25, 175], [0.2, 1]로 두지 않고 [0,200], [0,1]로 하고 제출시도를 하여서 여러시도의 점수를 알 수 없었던 것이 한계점



THANK YOU