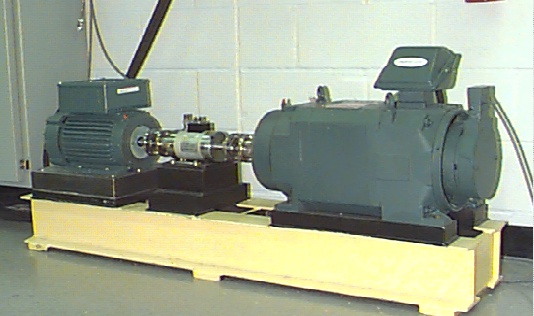
**파형 분석을 이용한 베어링 결함 진단**

1. **데이터 수집**
   1. 베어링 데이터 수집
   2. 데이터 분석
2. **웨이블릿 변환을 이용한 결함 진단**
   1. 샘플 데이터 생성
   2. CNN 학습
   3. CNN + 랜덤포레스트 학습
3. **Raw\_data와 Feature 생성을 이용한 결함 진단**
   1. 샘플 데이터 생성
   2. CNN + DNN 모델 학습
4. **데이터 수집**

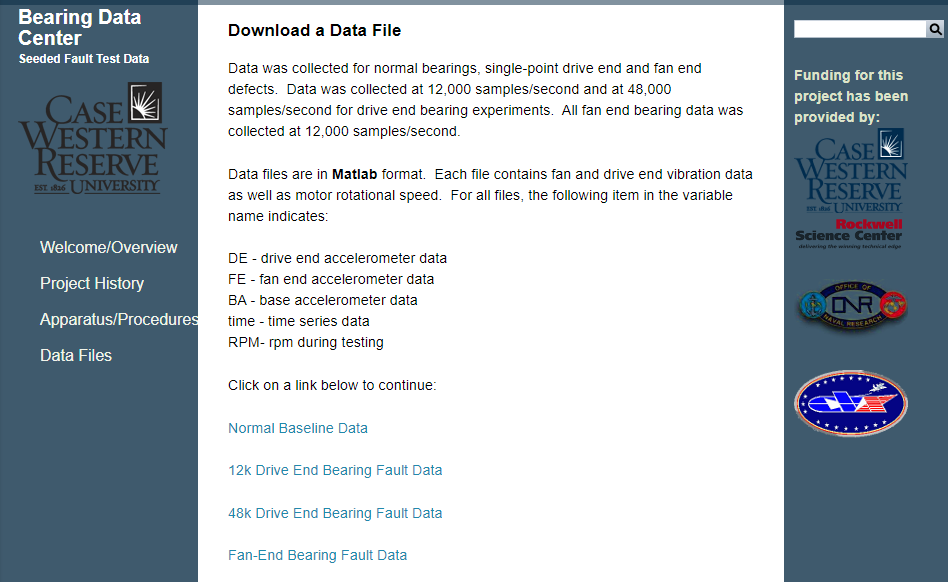
**A. 베어링 데이터 수집**

CWRU(Case Western Reserve University) Bearing Data Center 데이터 이용

https://csegroups.case.edu/bearingdatacenter/pages/welcome-case-western-reserve-university-bearing-data-center-website



데이터 수집을 위해 사용된 기기는 위 그림과 같이 구성되며 왼쪽이 2hp 모터, 가운데가 토크 변환기 / 인코더, 오른쪽이 동력계로 구성됨. 베어링은 모터 샤프트에 사용되며 결함의 크기는 7mils, 14mils, 21mils, 28mils, 40mils (1mil = 0.001인치)의 결함 직경을 갖도록 가공되어 있음. 결함은 위치에 따라 Inner, Ball, Outer로 나뉘며, 결함이 있는 베어링을 모터에 설치하고 걸리는 부하(모터 속도를 1797 ~ 1720RPM)에 따라 0~3으로 데이터를 기록하였음



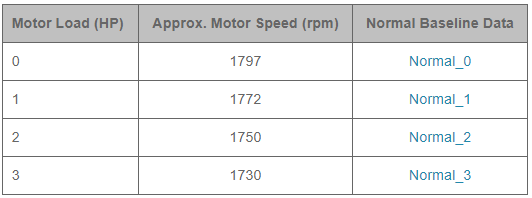
데이터는 12kHz로 저장된 Drive End Bearing Fault Data와 48kHz로 저장된 Drive End Bearing Fault Data, Normal BaseLine Data, 12kHz로 저장된 Fan-End Bearing Fault Data가 존재하며 Mat 파일로 제공됨.

데이터는 DE, FE, BA, Time, RPM 변수를 가지고 있으며 각 의미는 다음표와 같음

|  |  |
| --- | --- |
| 항목 | 설명 |
| DE | Drive end accelerometer data |
| FE | Fan end accelerometer data |
| BA | Base accelerometer data |
| Time | Time series data |
| RPM | Rpm during testing |

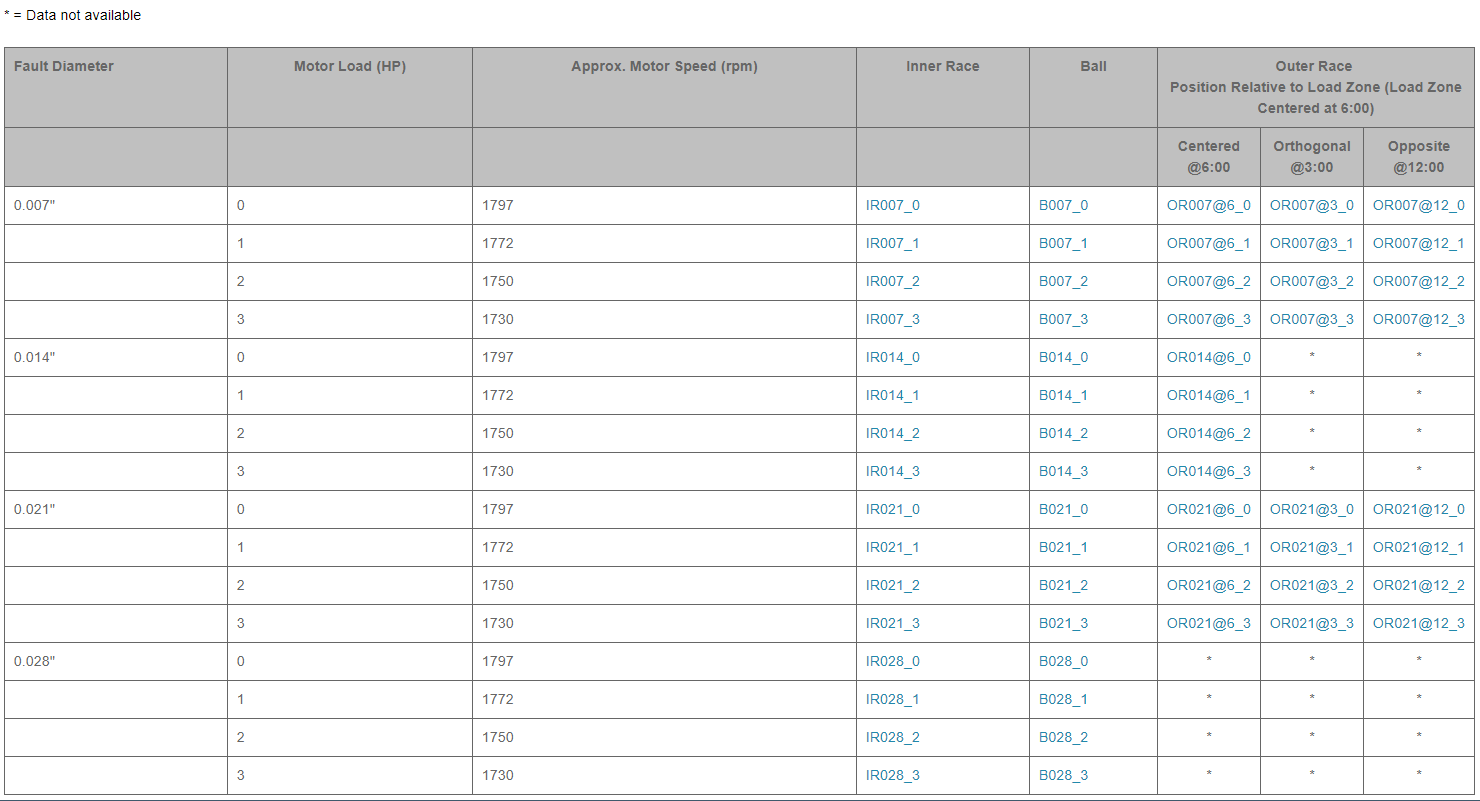
결함을 진단하기 위해서 사용한 데이터는 Normal BaseLine Data와 12kHz로 저장된 Drive End Bearning Fault Data를 이용

Normal Baseline Data는 정상 상태의 Bearing 진동 데이터로 0~3까지 4개의 부하별로 저장되어 있음

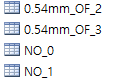


12k Drive End Bearing Fault Data는 12kHz로 저장된 데이터로 결함의 크기(0.007인치, 0.014인치, 0.021인치, 0.028인치)별로 나뉘어져 있으며 여기서 다시 0~3까지의 부하별로 데이터가 나뉨.

나뉜 데이터는 다시 결함의 위치(Inner, Ball, Outer)에 따라 데이터가 따로 존재하며 Outer 결함의 경우 bearing의 위치에 따라 6시 방향을 기준으로 직교인 3시방향, 맞은편인 12시 방향의 데이터로 나뉘어짐. 그러나 6시 방향의 데이터 외에는 결측치가 존재하기 때문에 6시 방향의 데이터만 이용하며, 0.028인치 결함 데이터의 경우 Outer 결함이 결측치이기 때문에 결측치가 없는 크기의 결함 데이터만 사용함



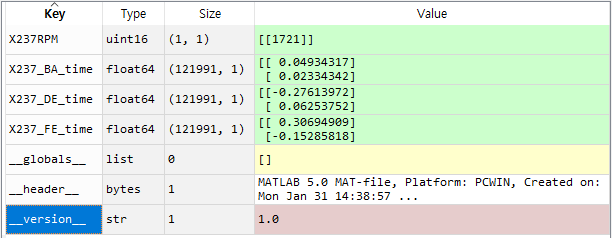
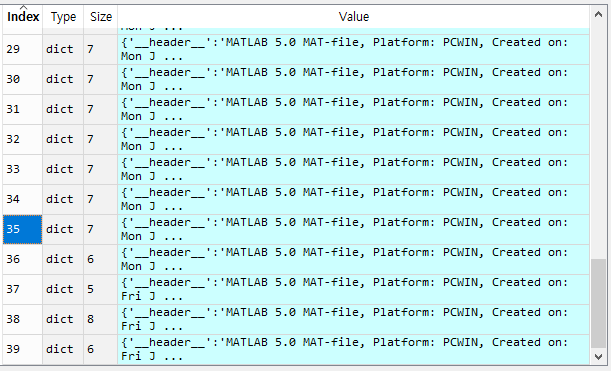
데이터는 mat형식으로 다운로드 되며 파일이름은 숫자로 저장되기 때문에 이름을 변경하여 데이터를 구분할 수 있도록 할 필요가 있음

결함의 크기, 상태, 부하 순으로 이름을 수정하였음,

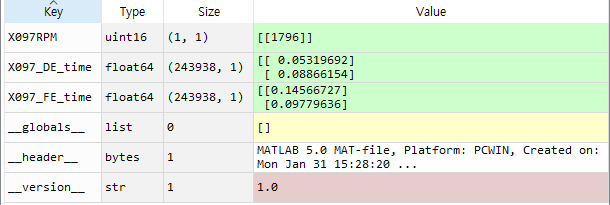
**B. 데이터 분석**

데이터는 총 40개이며 결함 데이터는 RPM, BA, DE, FE 데이터를 가지고 있음

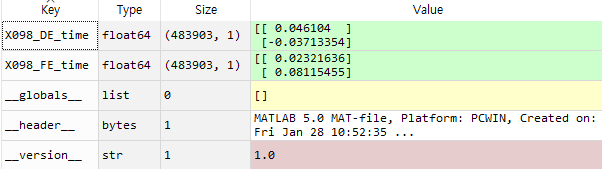


정상 데이터의 경우 결측치와 이상치가 존재하기 때문에 확인이 필요함

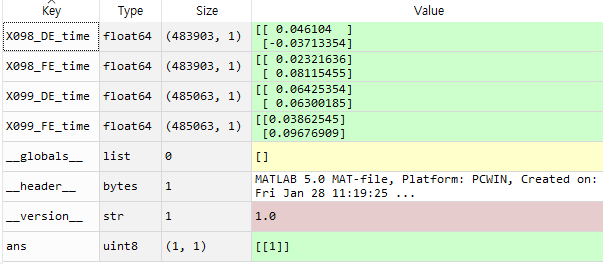
부하가 0인 정상상태 데이터에는 BA 데이터 결측



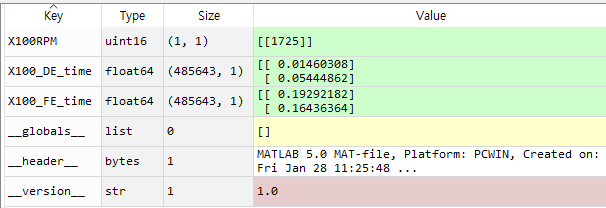
부하가 1인 정상상태 데이터에는 BA, RPM이 결측



부하가 2인 정상상태 데이터에는 부하가 1인 데이터와 섞여 있으며 ans라는 이상데이터와 BA가 결측

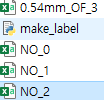
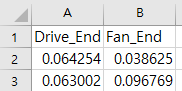
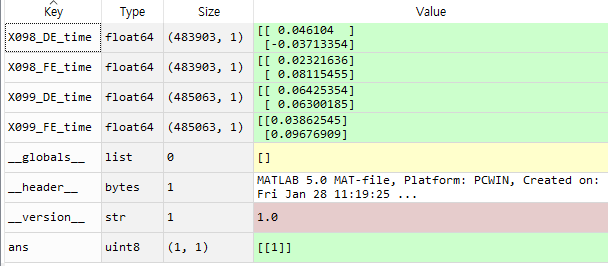


부하가 3인 정상상태 데이터에는 BA가 결측되어 있음

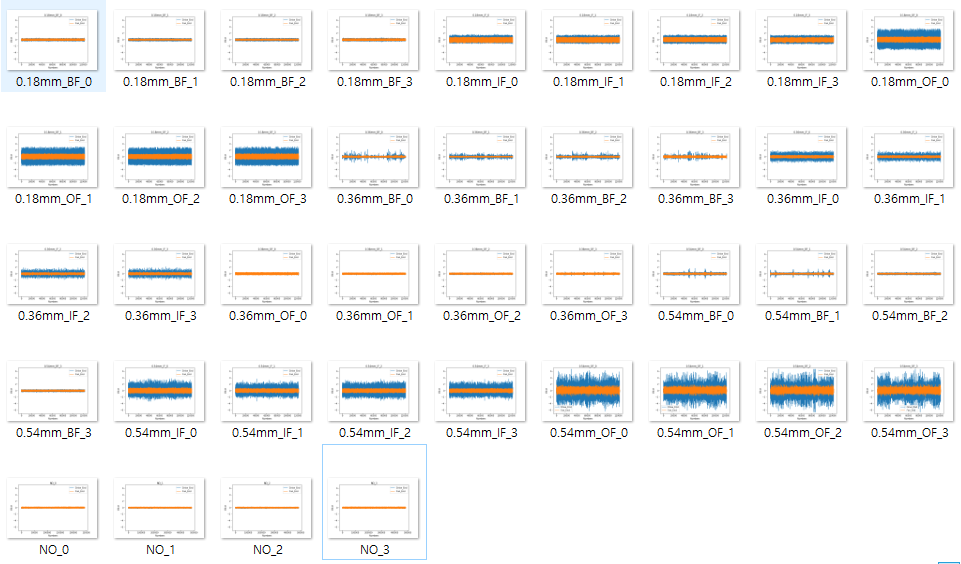


데이터를 csv형식으로 변환하여 저장하는 과정에서 RPM, BA 데이터는 결측치가 존재하기 때문에 DE, FE데이터만을 추출하여 Drive\_End, Fan\_End라고 변환하여 저장. 부하가 2인 정상상태 데이터의 경우 결측치 뿐만 아니라 잘못 들어간 데이터도 빼주기 위해 따로 작업을 하였음. 정상상태를 제외하고 나머지 결함데이터는 약 12만개 이며, 정상상태가 0인 경우 24만개, 나머지 정상상태 데이터는 48만개임

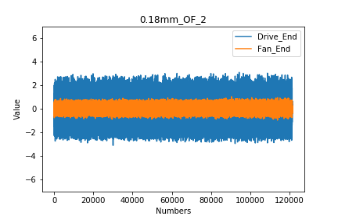
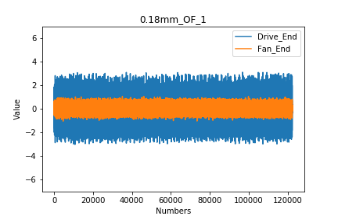
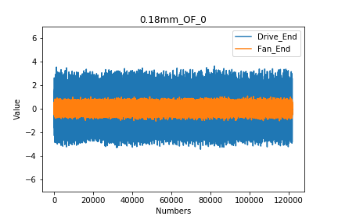
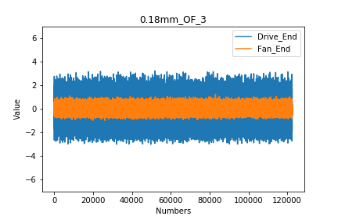
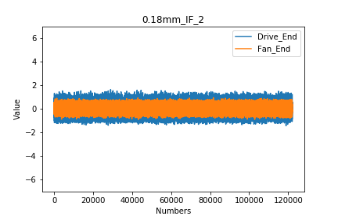
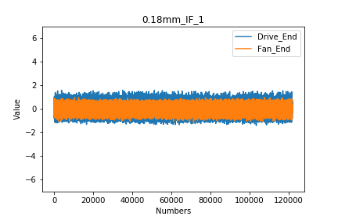
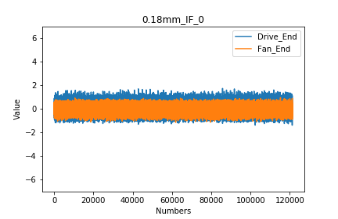
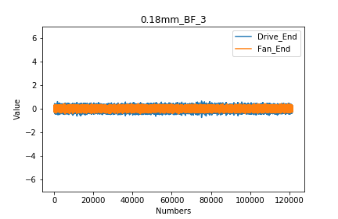
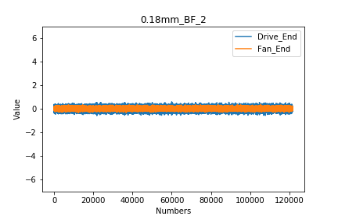
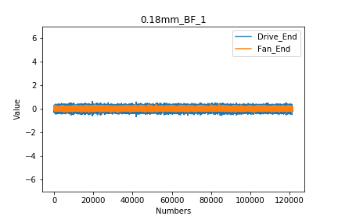
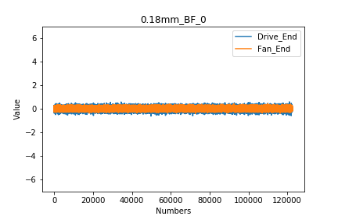
변환된 csv파일의 이름은 결함 데이터의 경우 결함의 크기, 결함의 위치, 부하의 크기로 저장하였으며, 정상 데이터의 경우 정상을 뜻하는 NO, 부하의 크기로 저장하였음



각 데이터 별로 Drive\_End와 Fan\_End데이터를 plot 그래프 형태로 시각화 하여 어떤 데이터가 결함 진단에 유의미할지 분석을 진행

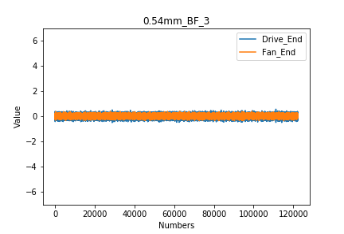
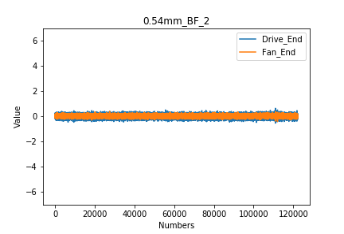
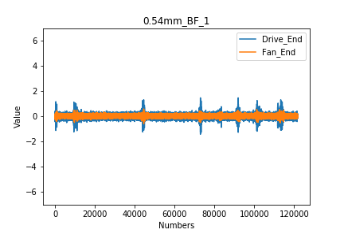
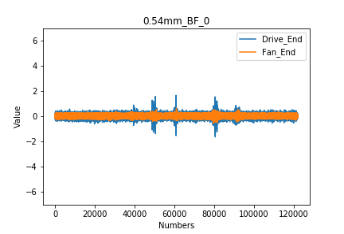
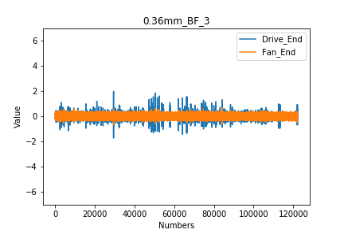
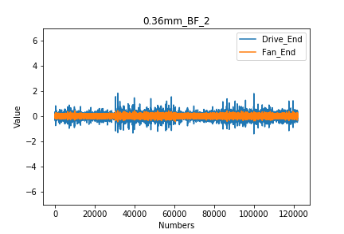
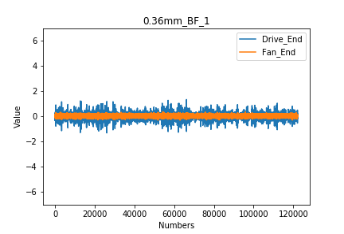
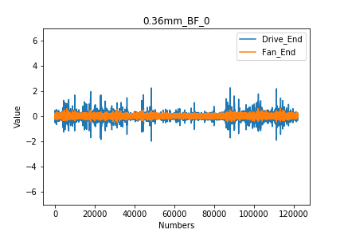
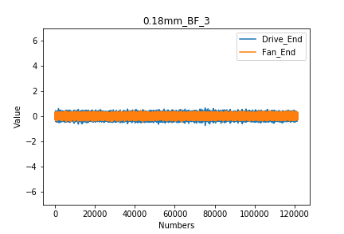
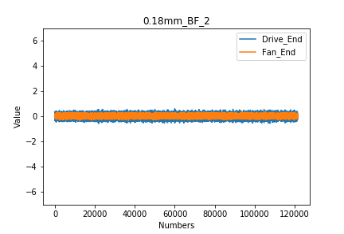
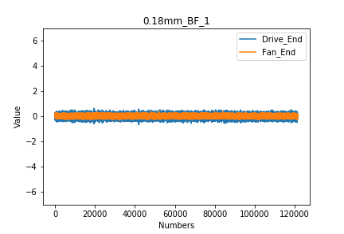
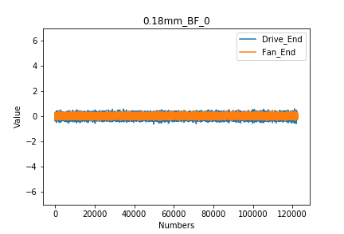


다음 그림은 결함의 크기가 0.007인치인 경우의 데이터를 분석한 결과로 가로에는 결함의 위치가 같으면서 부하가 다른 그래프들을 그려보았고, 세로는 결함의 위치에 따라 그래프를 그려 비교하였음.



파란색은 Drive\_End 데이터를 의미하며 주황색은 Fan\_End 데이터를 의미함. 분석결과 Fan\_End 데이터는 결함의 위치, 부하에 관계없이 2미만의 분포를 보이는 반면 Drive\_End 데이터는 부하에 따라 차이는 없지만 결함의 위치에 따라 큰 차이를 보임

다음 그림은 결함의 크기가 다른 경우의 데이터를 분석한 결과로 BF데이터를 이용해서 분석하였음. 가로에는 결함의 크기가 같으면서 부하가 다른 경우의 그래프를 그려보았고, 세로는 결함의 크기에 따라 그래프를 그려 비교하였음



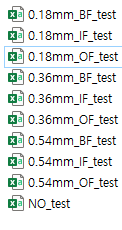
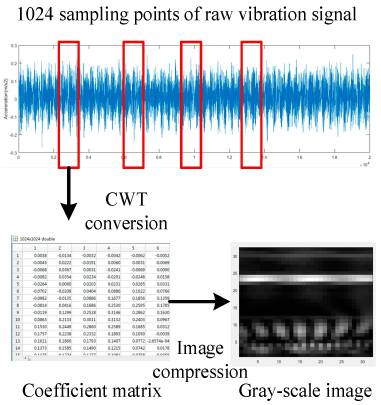
마찬가지로 파란색은 Drive\_End 데이터, 주황색은 Fan\_End데이터를 의미함. 이전의 분석 결과와 마찬가지로 Fan\_End 데이터는 큰 차이를 보이지 않음. Drive\_End 데이터는 결함의 크기가 0.007 인치인 경우 부하가 달라도 큰 차이를 보이지 않음. 0.014 인치와 0.021인치의 그래프에서 Drive\_End 데이터는 부하에 따라 차이를 보이긴 하지만 부하에 따른 명확한 차이가 보이지는 않음. 결함의 크기에 따른 차이를 살펴보면 부하가 낮은 경우에는 결함의 크기에 따라 명확한 차이를 보이지만 부하가 큰 경우에는 차이가 보이지 않는 경우가 있음

**2. 웨이블릿 변환을 이용한 결함 진단**

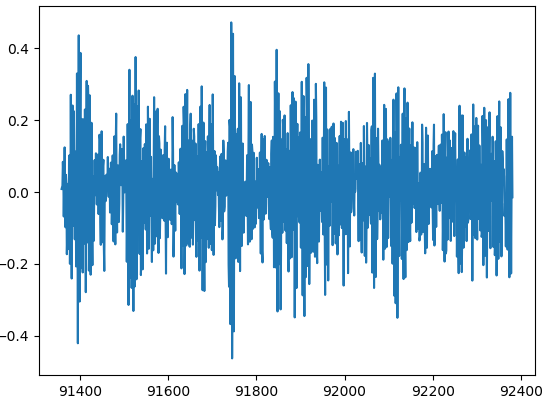
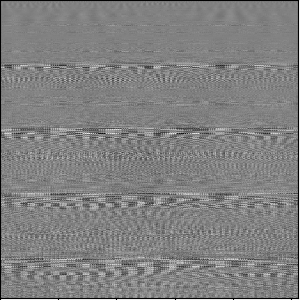
**A. 샘플 데이터 생성**

Fan\_End 데이터의 경우 결함의 종류를 분류하는데 무의미 하다 판단되어 Drive\_End 데이터만을 이용하여 결함을 진단.

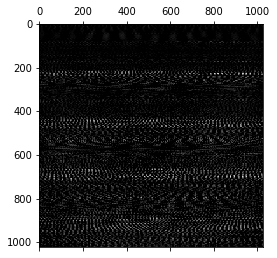
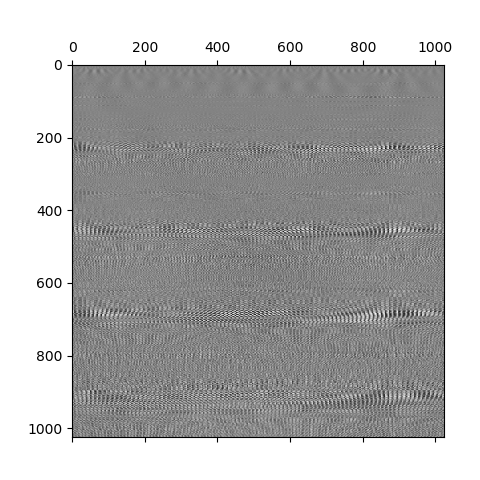
각 데이터 셋 별로 1024개의 포인트를 랜덤하게 추출하여 50개의 샘플 데이터 셋을 생성함. 이전의 분석에서 부하가 다른 경우 뚜렷한 차이를 보이지 않았기 때문에 결함의 크기, 결함의 위치에 따라 데이터를 통합하여 정상 데이터를 포함하여 각 400개의 샘플 데이터를 가진 10개의 샘플 데이터 셋 생성



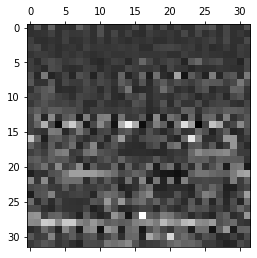
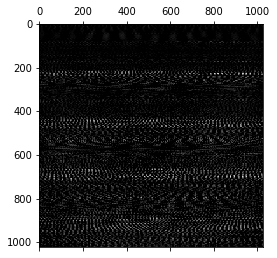
샘플 데이터를 그대로 사용하지 않고 진동 신호 분석에서 우수한 성능을 보인다고 알려진 Morlet 웨이블릿 변환을 통해 이미지로 변환함. 웨이블릿 변환은 Pywavlet을 이용하여 진행.

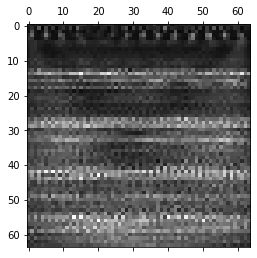
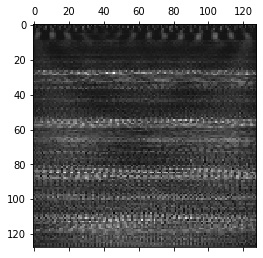
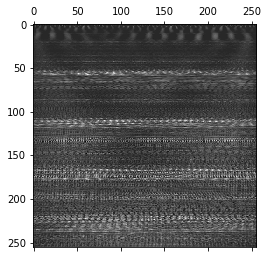
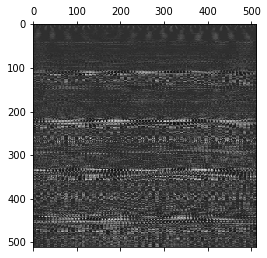
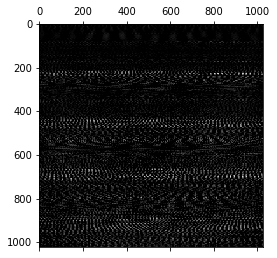
변환된 이미지의 특징을 뚜렷하게 하기위해 0이하의 값들은 0으로 만듬



이미지의 크기가 1024x1024로 CNN 모델에 사용하기에는 너무 크기 때문에 32x32로 Bicubic 방식을 이용해 축소 시킴. 이때, openCV에서 제공하는 bicubic으로 이미지를 한번에 줄일 경우 이미지의 특징이 모호해짐

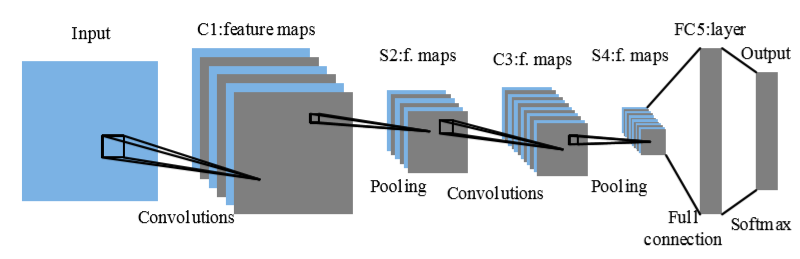


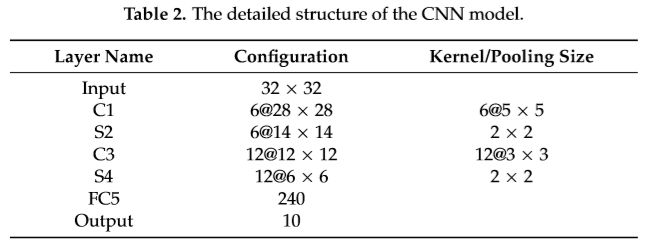
한번에 줄이지 않고 1/2로 조금씩 줄일 경우 특징을 더 명확하게 살려서 축소 시킬 수 있음



B. CNN 모델 학습

변환된 이미지를 이용해 10가지 상태를 구분하는 CNN 모델을 학습. 모델의 구조는 다음 그림과 같음



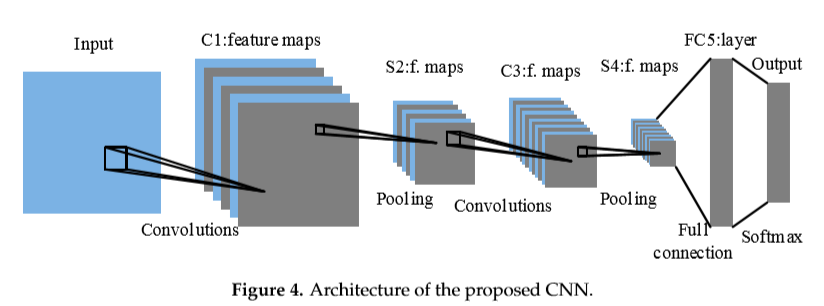


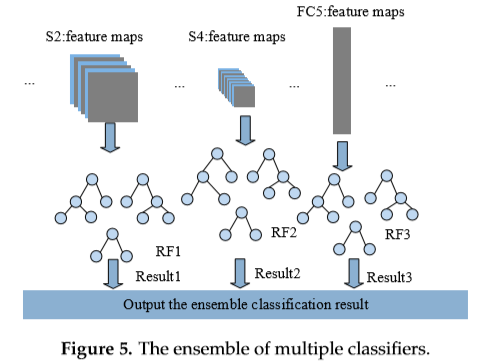
Learning\_rate는 0.000002로 60번의 epoch동안 Adam optimizer를 이용하여 학습을 진행한 결과 성능도 좋지 못하고, 학습할 때 마다 정확도의 변동이 큰 매우 불안정한 성능을 보임



C. CNN + 랜덤포레스트 모델 학습

모델의 성능을 개선하고, 안정화 시키기 위해 랜덤포레스트 앙상블을 이용함. 앙상블 모델의 구조는 다음 그림과 같음

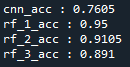
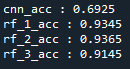




3개의 랜덤포레스트 모델을 만들고 이전의CNN에서 S2레이어에서 추출된 feature map과 S4레이어에서 추출된 feature map, FC5레이어에서 추출된 feature map을 이용하여 학습을 진행

랜덤포레스트의 입력데이터는 2차원 이하의 데이터만 사용가능하기 때문에 feature map을 2차원으로 변환하여 학습을 진행

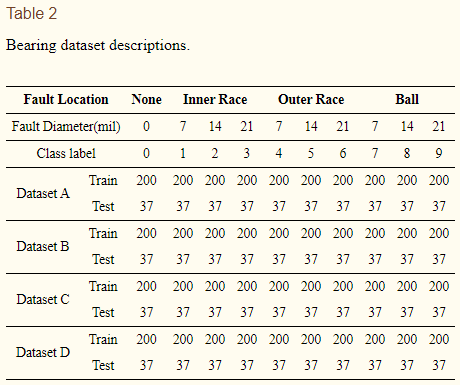
학습결과 성능개선도 이뤄졌으며 모델 안정화도 성공적으로 이워졌음

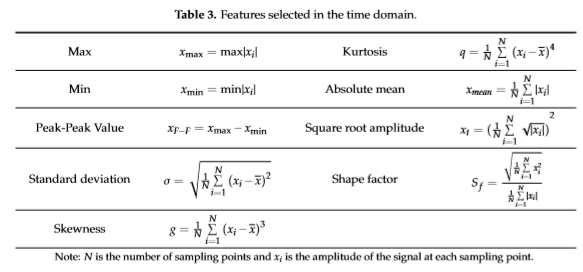
**3. Raw data와 feature 생성을 이용한 결함 진단**

A. 샘플 데이터 생성

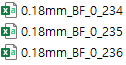
Raw data를 이용한 cnn 모델과 raw data를 이용해 9개의 feature를 생성하여 dnn 모델 입력으로 사용하여 두 모델을 합쳐서 결함을 진단. 사용하는 데이터는 이전과 같이 Drive\_End 데이터를 사용하며, 차이점으로는 가중치별로 데이터 셋을 A, B, C, D로 나누어서 결과를 비교하였음. 각 데이터 셋은 결함의 크기, 결함의 위치에 따라 10개의 상태를 가지며 각 상태 별로 랜덤하게512포인트를 추출하여 237개의 샘플 데이터 셋을 생성하였음. 각 상태별로 200개 총 2000개의 데이터가 학습에 사용되며 370개는 테스트에 이용함.



Raw data를 이용하여 생성된 9개의 feature는 다음과 같음



[Raw\_data 분류]

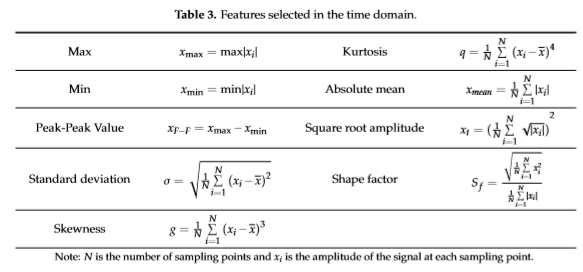


●●●

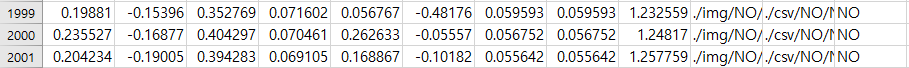
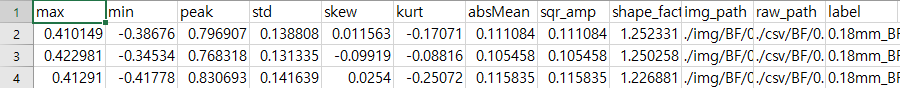
[Raw\_data를 이용한 feautre 생성]



●●●

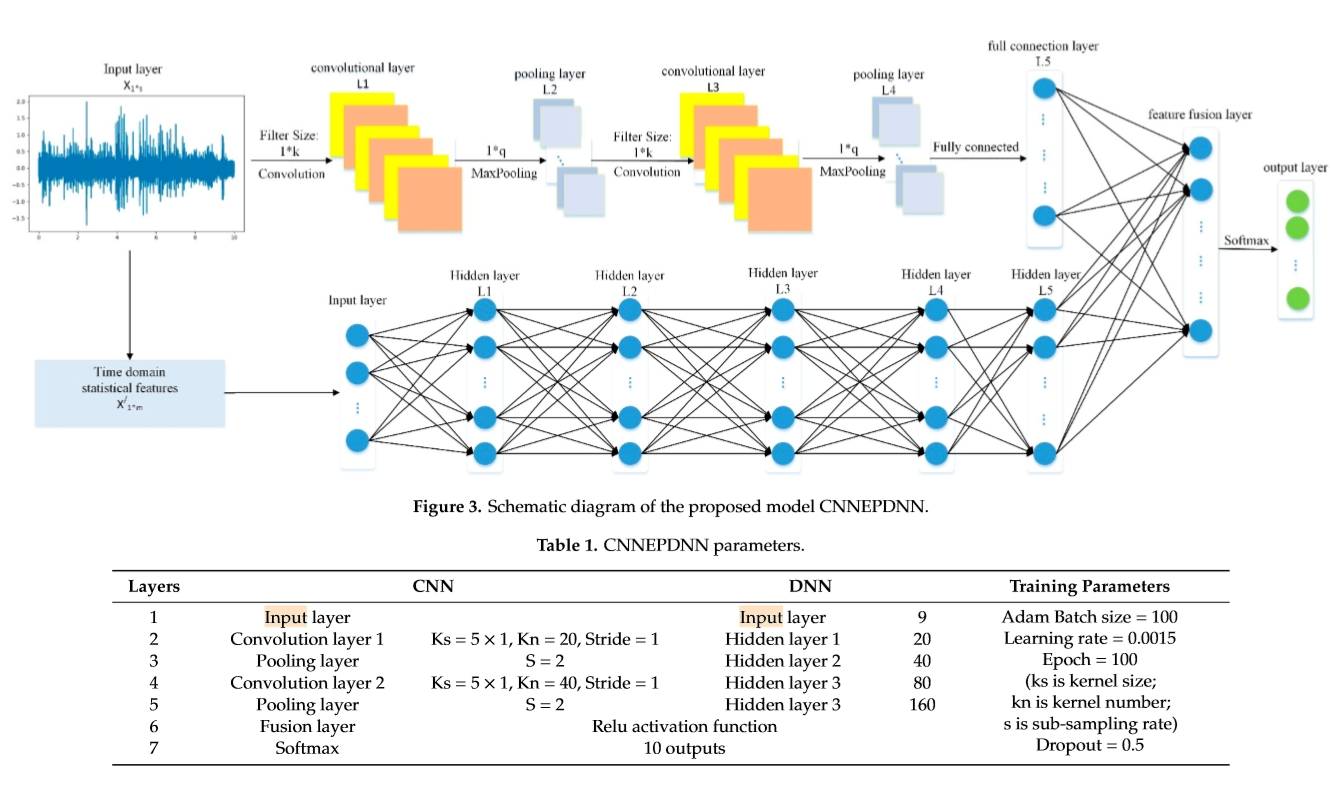


●●●



**B. CNN + DNN 모델 학습**

사용된 모델의 구조는 다음 그림과 같음. 그림에서 cnn 모델의 입력으로 2차원 이미지가 들어간 것처럼 보이지만 실제로는 1차원의 raw data를 그대로 입력하였음.



학습 결과 웨이블릿 변환과 랜덤포레스트를 이용한 모델보다 높은 성능을 보임

