


1차년도 주요 결과물

(과제명) 대규모 분산 에너지 저장장치 인프라의 안전한 자율운영
및 성능 평가를 위한 지능형 SW 프레임워크 개발
(과제번호) 2021-0-00077

- 결과물명 : 데이터 전처리 알고리즘(SW)
- 작성일자 : 2021년 12월 30일

과학기술정보통신부 SW컴퓨팅산업원천기술개발사업
“1차년도 주요 결과물” 로 제출합니다.

수행기관	성명/직위	확인
서울대학교 산학협력단	강명주/교수	

정보통신기획평가원장 귀하

<목차>

1. 개요	1
가. 목적	1
나. 범위	1
2. 기법	2
3. 소프트웨어	2
가. 주요 기능	2
나. 사용 환경	4
다. 사용 방법	4
라. 코드	4

1. 개요

가. 목적

- 본 문서는 “대규모 분산 에너지 저장장치 인프라의 안전한 자율운영 및 성능 평가를 위한 지능형 SW 프레임워크 개발” 사업의 주요 결과물에 대한 보고서이다.
- 안전 AI 분석 엔진 중 “데이터 전처리 알고리즘” 소프트웨어에 대한 설명을 한다.

[안전 AI 분석 엔진 SW]

구분	1차년도	2차년도	3차년도	4차년도	5차년도
안전 AI 분석 엔진	물성 진단 지표 후보(SW)	실험 데이터로 학습한 이상 탐지 알고리즘(SW)	전이학습을 완료한 이상 탐지 알고리즘(SW)	최적화된 이상 탐지, 성능 진단 알고리즘(SW)	이상 탐지 알고리즘 및 사고 대응 알고리즘(SW)
	데이터 전처리 알고리즘(SW)	실험 데이터로 학습한 성능 진단 알고리즘(SW)	전이 학습을 완료한 성능 진단 알고리즘(SW)	개선된 이상 탐지, 성능 진단 알고리즘(SW)	상태 진단 알고리즘 및 효율적 운영 분석 결과(SW)

나. 범위

- 배터리 데이터의 전처리 기능 설명
- 이를 구현하는 소프트웨어를 개발
- 구현 소프트웨어의 특징 설명

2. 기법

- SOC나 SOH와 같은 배터리의 지표를 추정하기 위하여 배터리의 충전과 방전 시의 전압과 전류 데이터가 사용되고 있음
- 본 프로그램은 이러한 배터리의 지표를 보다 정확하게 추정하기 위해 필요한 배터리 데이터의 전처리를 수행함
- 본 프로그램의 특징은 배터리 데이터의 수집장치 오류로 장시간 누락된 결측치의 적절한 대체값을 추정하고, 수치 보간법을 이용하여 데이터의 측정 시간 간격을 일정하게 보정하며, 다양한 지표의 수치 계산 과정에서 정상범위를 벗어나는 수치 불안정 문제를 수치계산 방식의 개선을 통하여 개선함.

3. 소프트웨어

가. 주요 기능

- 충방전 사이클 종류를 일반 사이클과 다양한 참조 사이클로 구분하고 용도별 다양한 인덱스 부여

```
# Define Ref_type (reference cycle type)
# A :
#   low current discharge at 0.04A
#   rest prior/post low current discharge
# B :
#   reference charge/discharge
#   rest post reference charge/discharge
# C :
#   pulsed load (rest/discharge)
#   rest post pulsed load
df.loc[:, 'ref_type'] = np.nan
df.loc[df['RW'], 'ref_type'] = 'none' # non-reference
df.loc[df['UNK'], 'ref_type'] = 'none' # non-reference
df.loc[df['charge_type'].str.contains('low'), 'ref_type'] = 'A'
df.loc[df['charge_type'].str.contains('reference'), 'ref_type'] = 'B'
df.loc[df['charge_type'].str.contains('pulsed'), 'ref_type'] = 'C'
```

```
# Define reference cycle number "ref_N"
# Note : Only ref cycles of type B are considered for the ref-cycle number.
# Note : if a cycle is non-reference or type A or type C, "ref_N" is -1.
df.loc[:, 'ref_N'] = np.nan
df.loc[df['RW'], 'ref_N'] = -1 # non-reference
df.loc[df['UNK'], 'ref_N'] = -1 # non-reference
df.loc[df['ref_type'] == 'A', 'ref_N'] = -1
df.loc[df['ref_type'] == 'C', 'ref_N'] = -1
ref_start = (df['ref_type'] == 'B') & (df['ref_type'].shift(1) != 'B')
df.loc[ref_start, 'ref_N'] = np.arange(1, ref_start.sum()+1)
df.loc[:, 'ref_N'] = df.loc[:, 'ref_N'].fillna(method='ffill')
```

- 데이터의 측정 시간 간격을 일정하게 보정

```
# Resampling timesteps -> interval : (exactly) 1 second
# Naively, assume that the experiment starts 1st, Jan, 2021. (doesn't matter)
df.index = df['t'] * 1e+9 + (1e+9 * 86400 * (365 * 51 + 13))
df.index = pd.DatetimeIndex(df.index)
df = df.resample('1S').first()
```

- 다양한 지표의 수치 계산 과정에서 정상범위를 벗어나는 수치 불안정 문제 보정

```
# 'Q' min/max clipping
df.loc[df['Q'] < 0, 'Q'] = 0.
df.loc[df['Q'] > df['capacity'], 'Q'] = df.loc[:, 'capacity']
```

- 장시간 결측치의 대체값 추정

```
# Time modification
df['dt'] = df['t'] - df['t'].shift(1)
df.loc[0, 'dt'] = 10
weird = np.array(df.loc[df['dt'] > 1000].index)
is_unknown_charge = df.loc[weird, 'V'].values > df.loc[weird-1, 'V'].values
unknown_charge = weird[np.where(is_unknown_charge)[0]]
unknown_discharge = weird[np.where(~is_unknown_charge)[0]]
df.loc[:, 'unknown_mark'] = np.nan
df.loc[unknown_charge-1, 'unknown_mark'] = 'unknown_charge_start'
df.loc[unknown_charge, 'unknown_mark'] = 'unknown_charge_end'
df.loc[unknown_discharge-1, 'unknown_mark'] = 'unknown_discharge_start'
df.loc[unknown_discharge, 'unknown_mark'] = 'unknown_discharge_end'
```

```
# copy 'Q' from no_unk_df to df
# (i.e, fill 'Q' in the 'unknown' interval using linear interpolation)
df.loc[df['UNK'] == False, 'Q'] = no_unk_df.loc[:, 'Q']
del no_unk_df
df.loc[:, 'Q'] = df.loc[:, 'Q'].interpolate() # In the 'unknown' interval, linearly interpolate 'Q'
```

나. 사용 환경

- 본 프로그램은 python 코드로 구현되었음
- Python이 설치된 어떠한 OS 환경에서도 사용이 가능함
- 코드의 구동에는 python의 OS, numpy, pandas, scipy 라이브러리가 필요함

다. 사용 방법

- 입력과 출력 데이터는 parquet 형식을 사용함
- data_file에 입력 데이터 path, basic_file과 advanced_file에 출력 데이터 path를 적어준 뒤
- python preprocessing.py 를 실행해줌

라. 코드

- OpenESS/SNU_AI/data_preprocess/preprocessing.py 코드 중 preprocess_basic 모듈