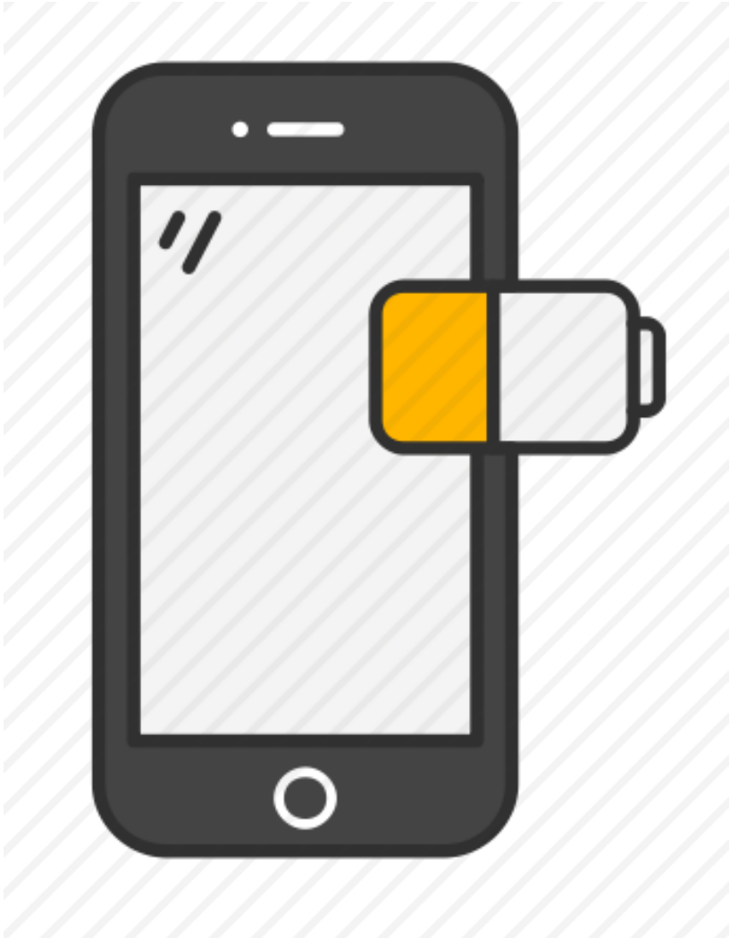
# PREDICTING USABLE LIFETIME OF LITHIUM-ION BATTERIES

Victor W. Hu, Yusong Liu, Rossana Scavone, Maitri Uppaluri

## What Is PULOLB?



Polymer lithium-ion pouch cell
($LiCoO_2$ | $C_6$)

PULOLB is a software package, developed in Python, that can predict the remaining usable lifetime of a lithium-ion battery.

Knowledge of the current state of a device's battery is important to users. To accurately assess these states, time-intensive procedures often have to be done. However, PULOLB can grant access to that information, in a shorter time frame.

Using long-term experimental, full-cycling data of the battery in question, the software is designed to allow a user to estimate the current state of their device's battery when the battery has been partially charged and discharged over a shallower depth of discharge.

## How Does The Package Work?

`predict_capacity:`

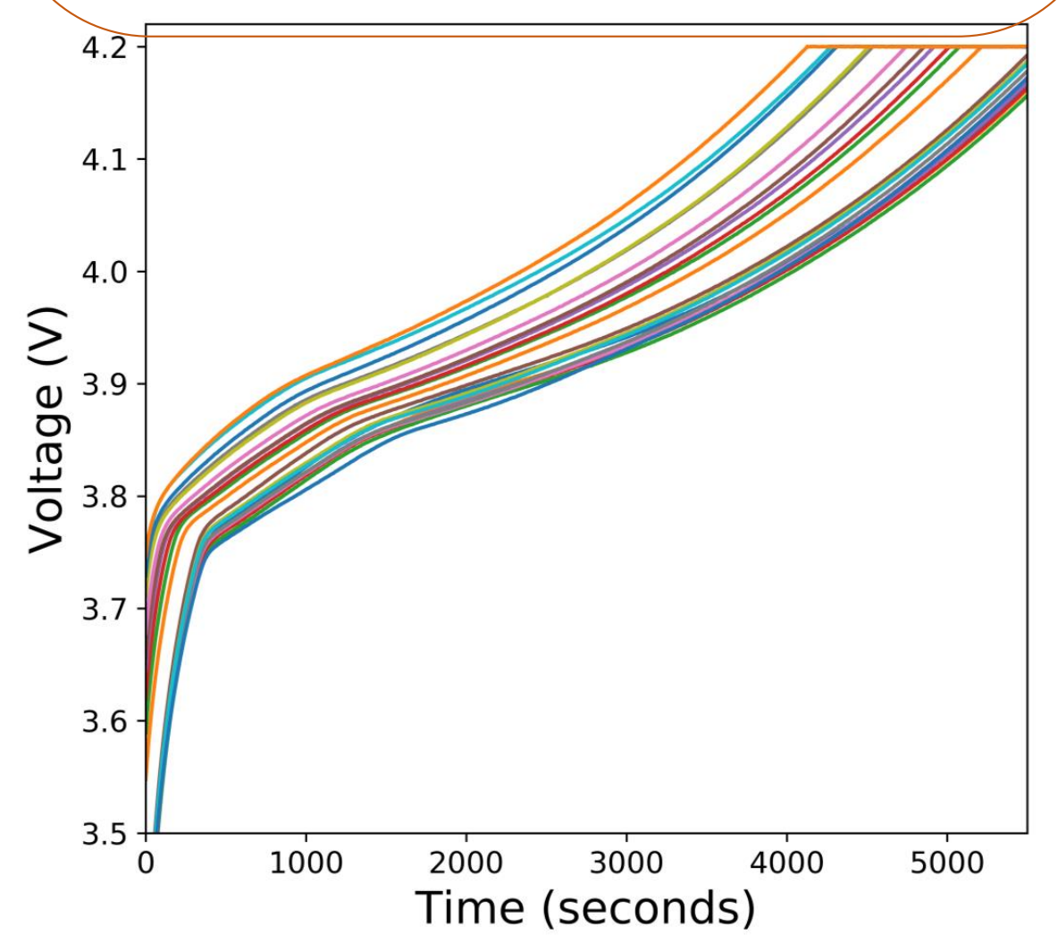
`distance_cycle_to_full:`

`partial_to_full:`

`get_capacity:`

`get_lifetime:`

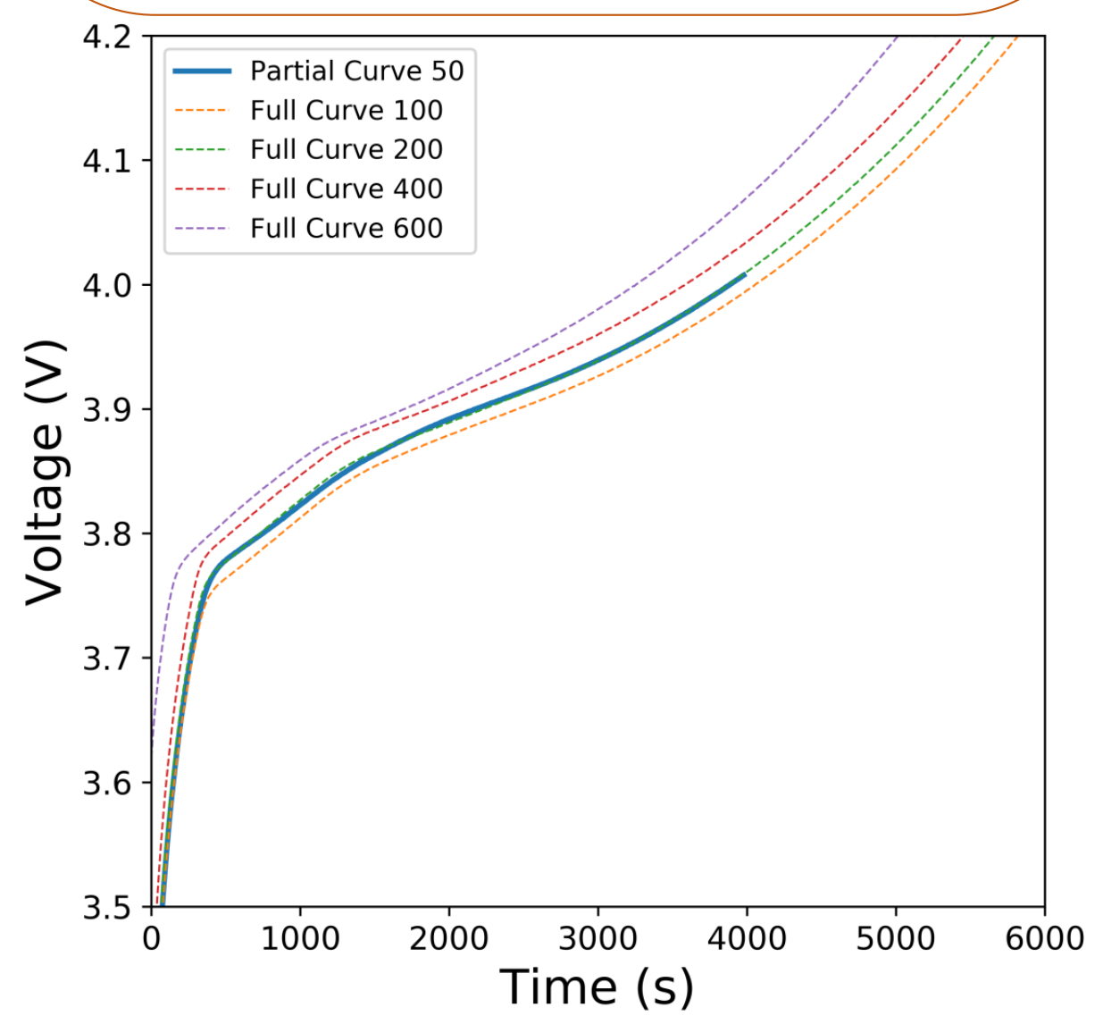
The **calce** PL samples can be used by the predict_capacity module to estimate the capacity and the lifetime of the user's battery based on a single partial cycle of the same battery and same C-cycling rates.



Full charge cycles of a pouch lithium ion battery at a constant C/2 rate. The cycles shown are from cycle 0 to cycle 900. It is evident that at high cycle numbers, the battery has lost appreciable capacity.

The distance between the two time series is calculated using **Dynamic Time-Warping** and with the **KNN algorithms**, the full curve that has the shortest Euclidean distance to the partial curve gives its cycle number.

The given set of **partial** charge or discharge curve is fitted into the library of **full** curves. For each partial curve, the algorithm predicts closest resembling full cycle curve and returns that cycle number.

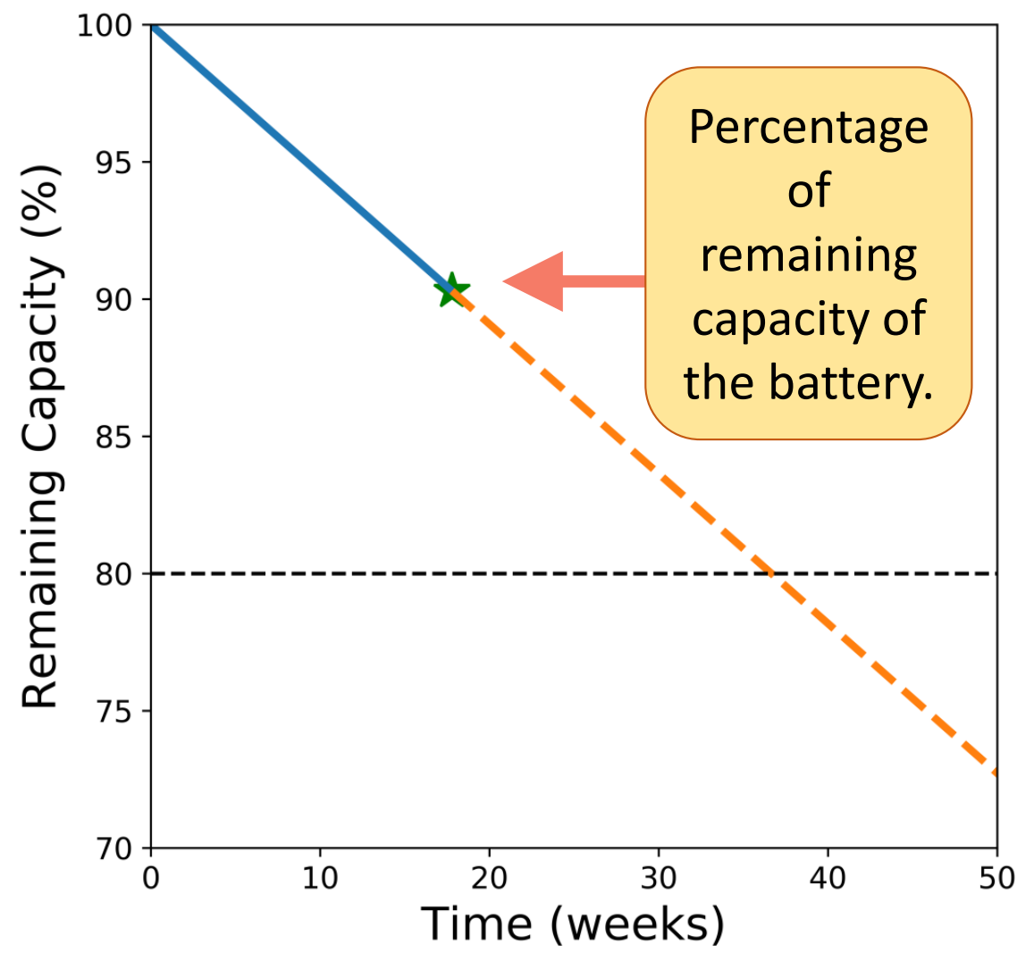


The partial curve, shown in a thicker line, is plotted against multiple full curves for comparison. Through the algorithm, it is expected this partial curve is most related to the state of the 200th full cycle.

Once the corresponding cycle number is predicted, the **capacity** of the battery is determined. The lifetime of the battery can be calculated from the capacity.

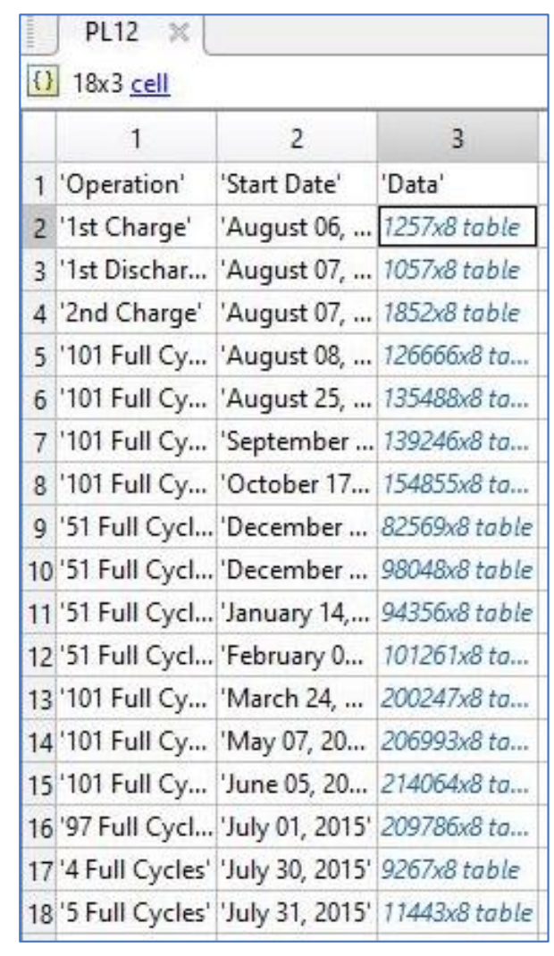
Prediction of the battery's lifetime.

The package returns the current battery status and predicts when the battery will reach 80 % of its rated capacity, a common metric of the **remaining lifetime,** based on the time of the first cycle and the cycle of interest.

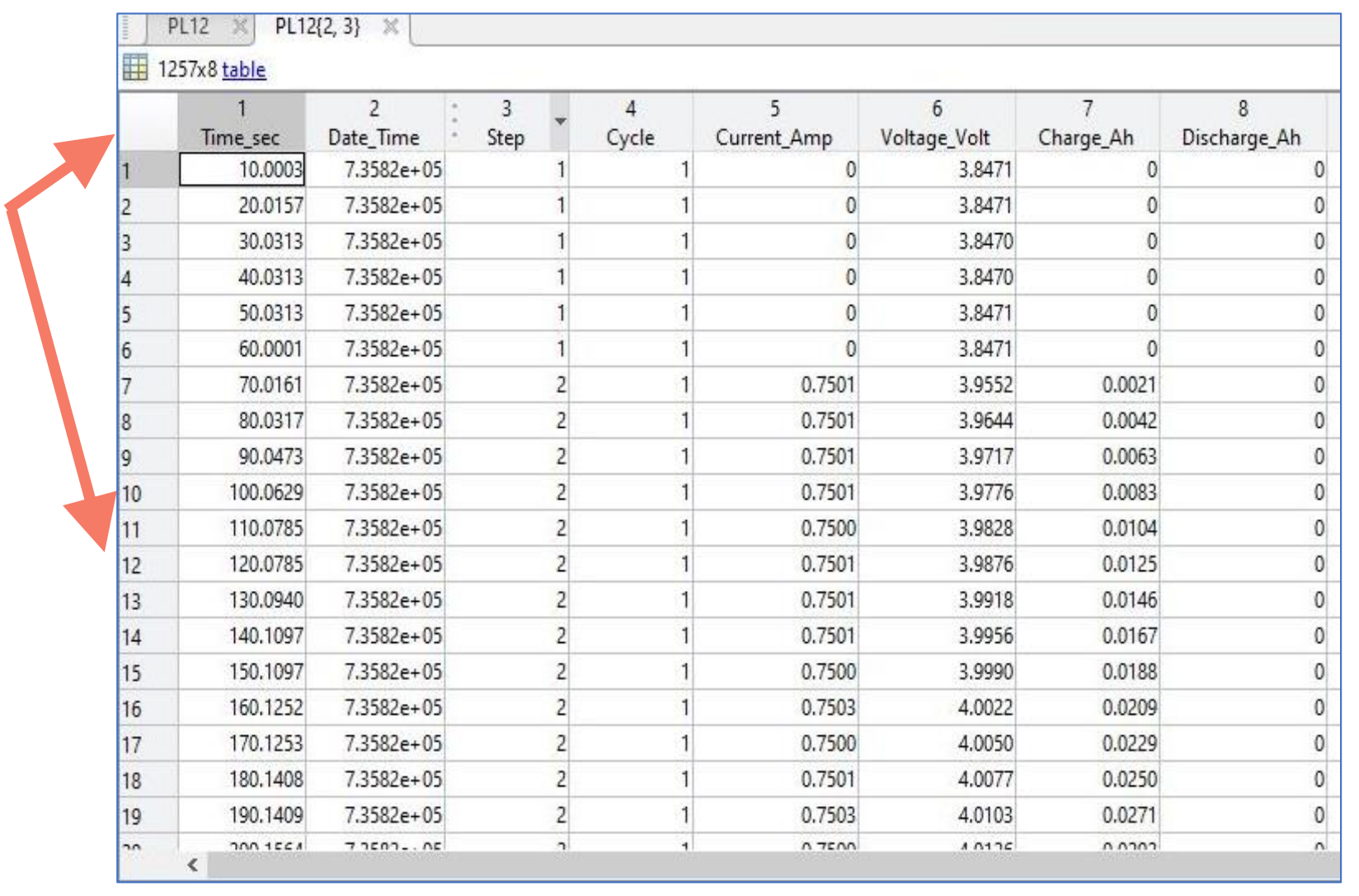


## Data Cleaning

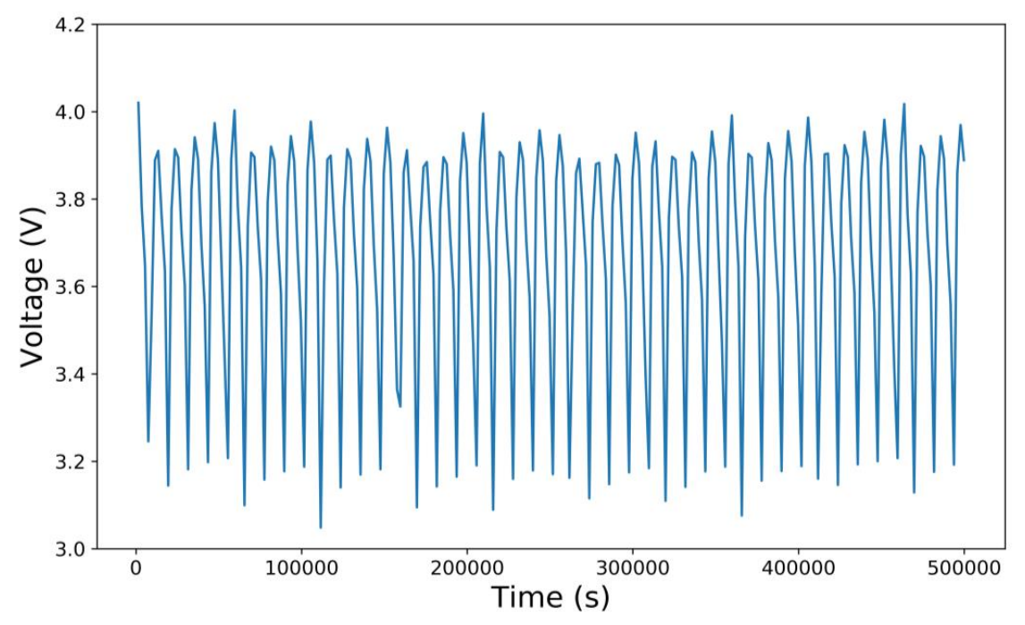
`import_data:`



| | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 'Operation' | 'Start Date' | 'Data' |
| 2 | '1st Charge' | 'August 06, ... | 1257x8 table |
| 3 | '1st Dischar... | 'August 07, ... | 1057x8 table |
| 4 | '2nd Charge' | 'August 07, ... | 1852x8 table |
| 5 | '101 Full Cy... | 'August 08, ... | 126666x8 ta... |
| 6 | '101 Full Cy... | 'August 25, ... | 135488x8 ta... |
| 7 | '101 Full Cy... | 'September ... | 139246x8 ta... |
| 8 | '101 Full Cy... | 'October 17... | 154855x8 ta... |
| 9 | '51 Full Cycl... | 'December ... | 82569x8 table |
| 10 | '51 Full Cycl... | 'December ... | 98048x8 table |
| 11 | '51 Full Cycl... | 'January 14,... | 94356x8 table |
| 12 | '51 Full Cycl... | 'February 0... | 101261x8 ta... |
| 13 | '101 Full Cy... | 'March 24, ... | 200247x8 ta... |
| 14 | '101 Full Cy... | 'May 07, 20... | 206993x8 ta... |
| 15 | '101 Full Cy... | 'June 05, 20... | 214064x8 ta... |
| 16 | '97 Full Cycl... | 'July 01, 2015' | 209786x8 ta... |
| 17 | '4 Full Cycles' | 'July 30, 2015' | 9267x8 table |
| 18 | '5 Full Cycles' | 'July 31, 2015' | 11443x8 table |



| | 1<br>Time_sec | 2<br>Date_Time | 3<br>Step | 4<br>Cycle | 5<br>Current_Amp | 6<br>Voltage_Volt | 7<br>Charge_Ah | 8<br>Discharge_Ah |
|---|---|---|---|---|---|---|---|---|
| 1 | 10.0003 | 7.3582e+05 | 1 | 1 | 0 | 3.8471 | 0 | 0 |
| 2 | 20.0157 | 7.3582e+05 | 1 | 1 | 0 | 3.8471 | 0 | 0 |
| 3 | 30.0313 | 7.3582e+05 | 1 | 1 | 0 | 3.8470 | 0 | 0 |
| 4 | 40.0313 | 7.3582e+05 | 1 | 1 | 0 | 3.8470 | 0 | 0 |
| 5 | 50.0313 | 7.3582e+05 | 1 | 1 | 0 | 3.8471 | 0 | 0 |
| 6 | 60.0001 | 7.3582e+05 | 1 | 1 | 0 | 3.8471 | 0 | 0 |
| 7 | 70.0161 | 7.3582e+05 | 2 | 1 | 0.7501 | 3.9552 | 0.0021 | 0 |
| 8 | 80.0317 | 7.3582e+05 | 2 | 1 | 0.7501 | 3.9644 | 0.0042 | 0 |
| 9 | 90.0473 | 7.3582e+05 | 2 | 1 | 0.7501 | 3.9717 | 0.0063 | 0 |
| 10 | 100.0629 | 7.3582e+05 | 2 | 1 | 0.7501 | 3.9776 | 0.0083 | 0 |
| 11 | 110.0785 | 7.3582e+05 | 2 | 1 | 0.7500 | 3.9828 | 0.0104 | 0 |
| 12 | 120.0785 | 7.3582e+05 | 2 | 1 | 0.7501 | 3.9876 | 0.0125 | 0 |
| 13 | 130.0940 | 7.3582e+05 | 2 | 1 | 0.7501 | 3.9918 | 0.0146 | 0 |
| 14 | 140.1097 | 7.3582e+05 | 2 | 1 | 0.7501 | 3.9956 | 0.0167 | 0 |
| 15 | 150.1097 | 7.3582e+05 | 2 | 1 | 0.7500 | 3.9990 | 0.0188 | 0 |
| 16 | 160.1252 | 7.3582e+05 | 2 | 1 | 0.7503 | 4.0022 | 0.0209 | 0 |
| 17 | 170.1253 | 7.3582e+05 | 2 | 1 | 0.7500 | 4.0050 | 0.0229 | 0 |
| 18 | 180.1408 | 7.3582e+05 | 2 | 1 | 0.7501 | 4.0077 | 0.0250 | 0 |
| 19 | 190.1409 | 7.3582e+05 | 2 | 1 | 0.7503 | 4.0103 | 0.0271 | 0 |



Partial cycling data (C/2 rate) of a PL sample from the CALCE database after the data has been cleaned and imported into Python

### Functions used to clean the data:

`import_data:`

`single_pd_matlab_data:`

Converts the .mat files into a single Pandas DataFrame, where all the data in the tables are unpacked, and the cycles are properly annotated

`sort_data:`

`by_cycle:`

Sorts the data in the pandas dataframe by cycle number, stores this data as a dictionary with the cycle number as the key.

`charge_discharge:`

Separates the charge and discharge data by their respective cycles.

## Packages Used To Develop The Machine Learning Code

### Packages used for data cleaning:

`hdf5storage:`

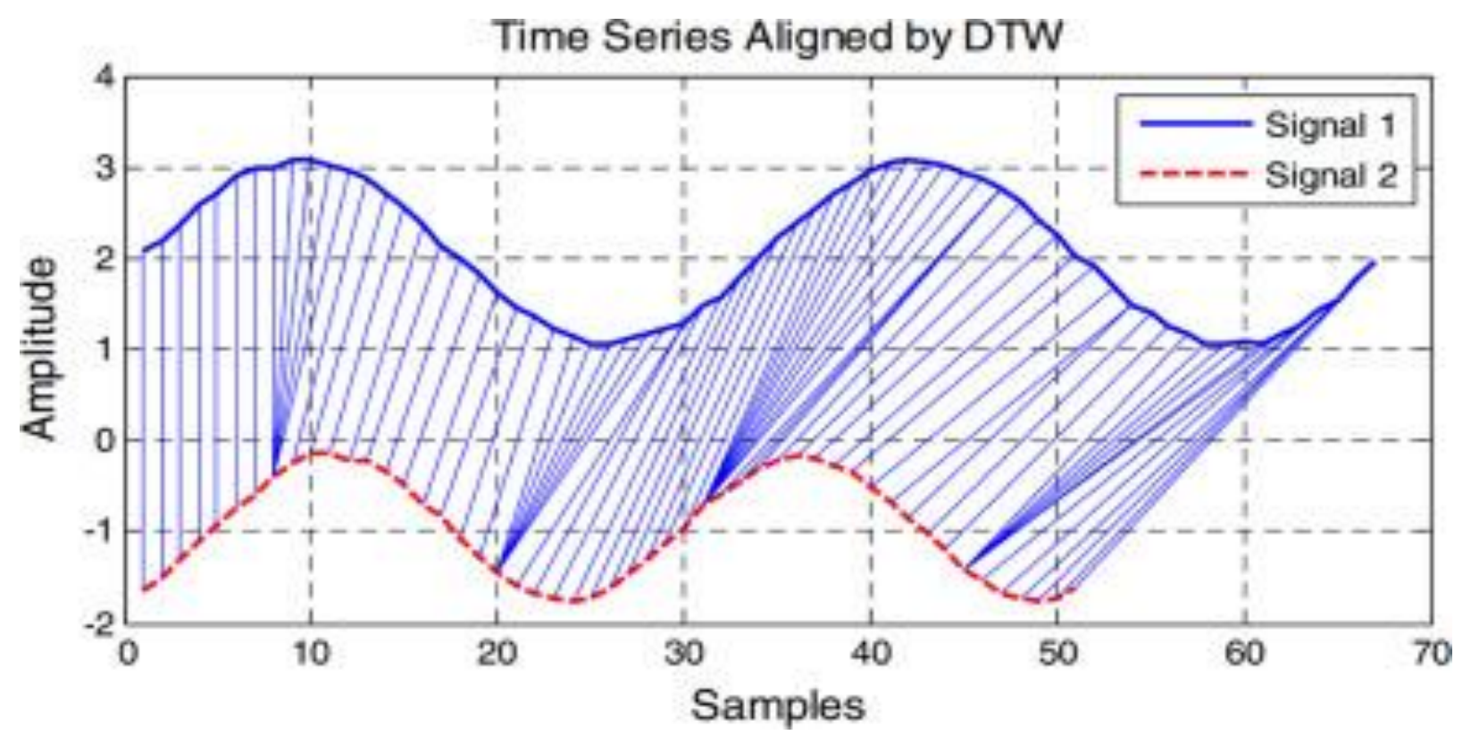
A package that can read/write HDF5 (Hierarchical Data Format) files. The data is classified into HDF5 groups, which contain HDF5 datasets that can be read as NumPy arrays. Each dataset is stored inside a dictionary.

`h5py:`

Required for full functionality of hdf5storage.

### Dynamic Time Warping

The distance between the two time series are calculated using **DTW**. For this project, the **Euclidean** distance between the two time series was calculated.



## Future Work

- Create a GUI for the user to input data more effectively.
- Expand the `import_data` function to allow different kinds of input files to be read and implemented.
- Improve speed of the code by trying other machine learning algorithms.
- Improve the accuracy of the cycle, capacity and lifetime prediction of the battery.
- Better methods to extrapolate data for a more accurate prediction of the battery's lifetime.

## References

Salvador, S. & Chan, P. FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space.

Lu, B., Xu, S., Stuber, J. & Edgar, T. Constrained selective dynamic time warping of trajectories in three dimensional batch data - ScienceDirect.

DIRECT
Data Intensive Research
Enabling Clean Technologies

NSF

CLEAN ENERGY
INSTITUTE
UNIVERSITY of WASHINGTON

calce® CENTER FOR ADVANCED LIFE CYCLE ENGINEERING

CHEMICAL ENGINEERING
UNIVERSITY of WASHINGTON