SoC Midterm Help Document

Denoising Transformer Current Data

Abstract:

This project revolves around enhancing the quality of transformer current data sampled every 5 minutes, sourced from a solar power generation site. The dataset likely contains fluctuations, anomalies, and missing values due to various factors such as weather conditions, equipment malfunctions, or measurement errors. Our objective is to identify 'good', 'bad', and 'missing' days within this dataset and improve the quality of the data by addressing inconsistencies and filling in missing values. Leveraging machine learning techniques, we aim to develop a model using data from the identified 'good' days to learn patterns and relationships, which will then be validated and utilized to enhance the quality of data from 'bad' and 'missing' days. Through this process, we intend to provide a more reliable dataset for analysis and decision-making in the context of solar power generation.

The Problem simplified :

Part 1

Classifying the data into three baskets namely : good , bad and missing. Days containing very high fluctuations/noise were deemed as bad.

Empty days had values of current very close to one throughout the day.

Part 2

Converting the good data into very good data by removing the outliers. By creating an idealised data set containing optimal values of current at any given time.

Part 3

Creating new features using feature engineering by splitting the timestamp into hours

,minutes, their higher powers and mutual products.

Continued...

Part 4

Applying Multiple Linear Regression along with Feature Elimination to identify the most relevant features out of 28 initial engineered features.

Part 5

Utilizing a Machine Learning (ML) model to predict the current values on the missing and unknown times

Part 6

Using plots to generate interactive graphs and using descriptive statistics to analyse the results.

Solutions

Solution 1

To identify the different classes of days we generated an ideal good day by taking the average value of current at every timestamp of a few excellent days which were clearly visible from our graphs.

Solution 2

Converted good days into very good by replacing the outliers by the created ideal day dataset thus further minimising the noise in the good days.

Solution 3

Created new features by using various powers of the minutes and hours value extracted from the timestamp. Also used feature elimination using p-value analysis to remove unwanted values.

Achievements

Achievement 1

Identified around 81 good days and 78 missing days whereas the remaining days mere the bad days.

Achievement 2

The R2 squared value on the tested data data was around 0.961 and the F - statistic value was around 1.09 *10^4.

Achievement 3

Reduced the error of good days from x to y , the errors of bad days from z to a and Finally reduced the overall error from b to c.

Solution

Clearly Explained !!

Fully Automated !!

0% Manual !!

Step 1:

Plotting The data to get a visual feel of the data

We first plotted all the data points and understood a few important concepts.

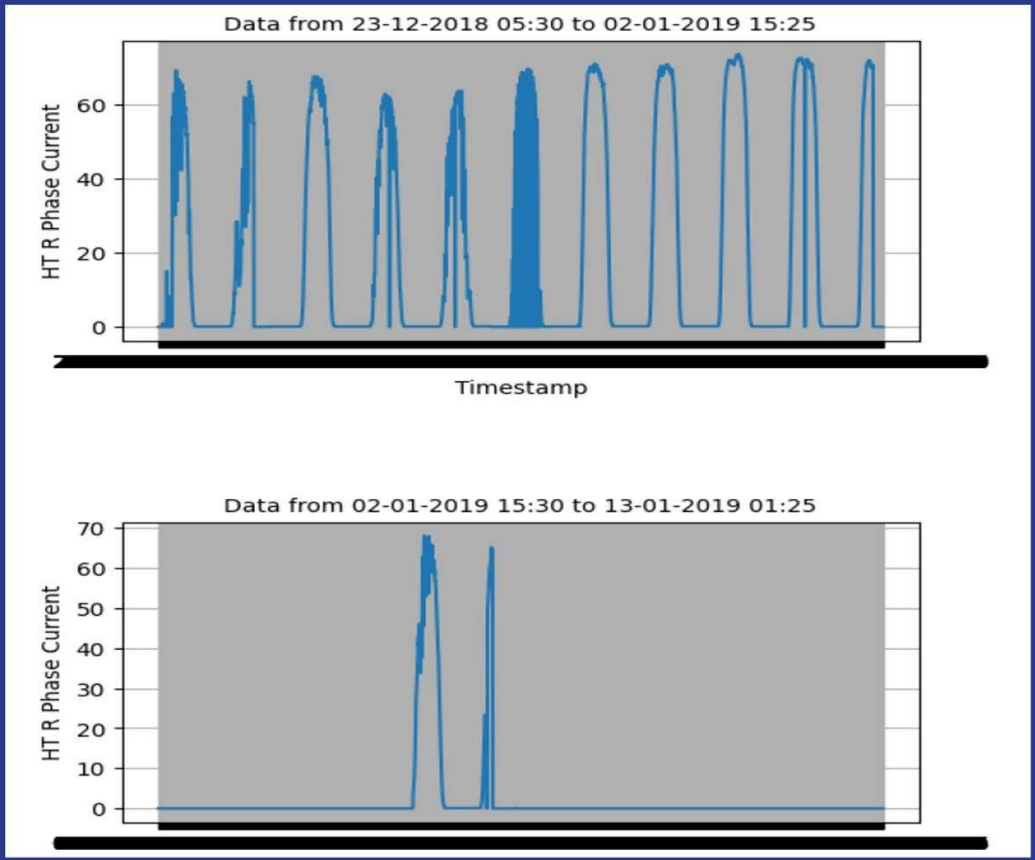
Firstly as the data is from a solar power plant so it is perfectly correct for the htr current values to be zero after sunset and before sunrise.

Also even in the good days the few outliers which are present occur as a bunch

.

Finally and perhaps the most important thing is that the data feels to be a half wave sinusoidal curve as is obtained as an output by a half rectifier.

Graphs :



Data from 23-12-2018 05:30 to 02-01-2019 15:25
HT R Phase Current
0
20
40
60
Timestamp
Data from 02-01-2019 15:30 to 13-01-2019 01:25
HT R Phase Current
0
10
20
30
40
50
60
70

Step 2:

Ideal Day Creation to Accurately Identify "Bad" and "Missing" Days

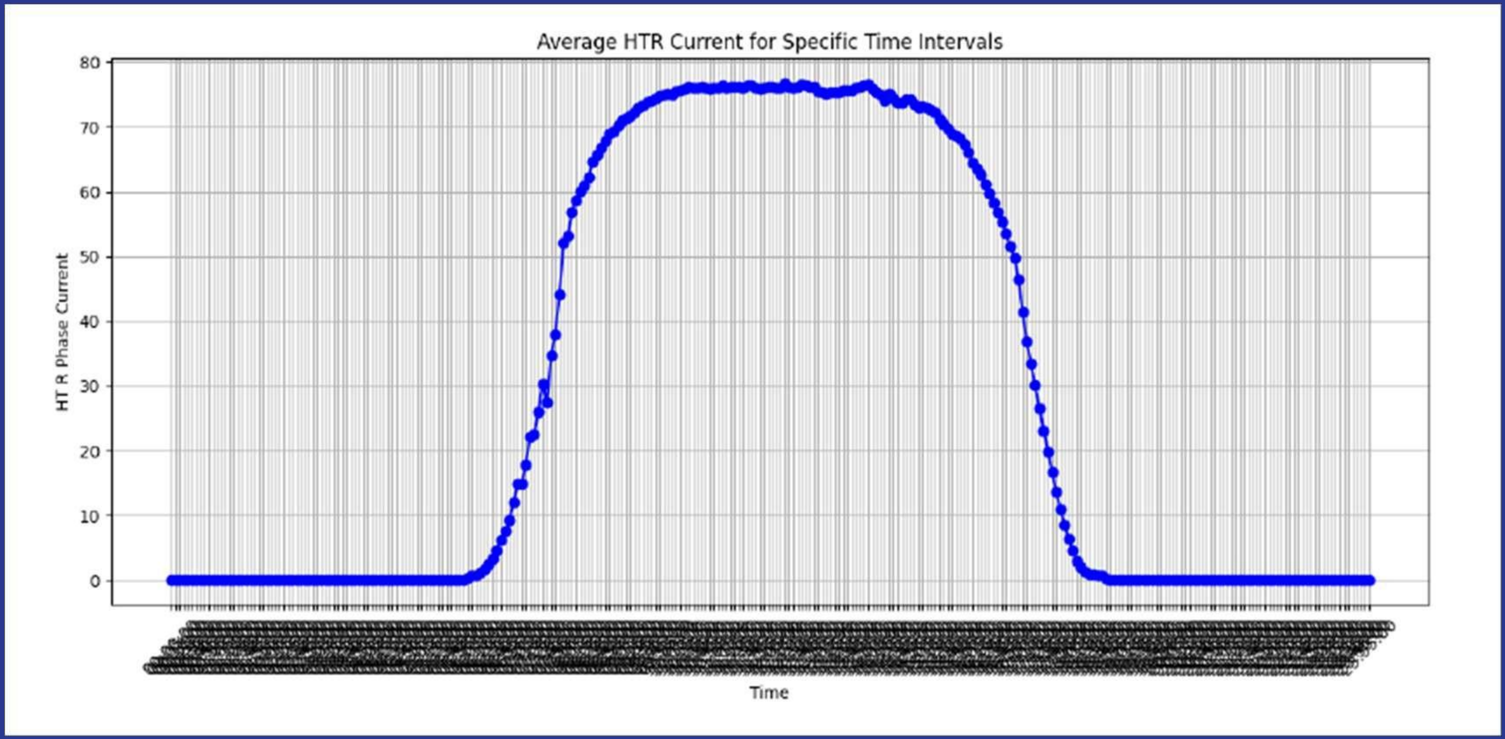
We created an ideal day by computing the average of all the best days at every time of the day.

We then calculated the root sum of square of differences between the ideal and the actual htr current values at each time for a given day.

We then checked the errors for the best days and then generalized the fact that if the errors are between a set values (close to the best days) then they were good and if the errors were close to highest or the highest they were missing days and all the remaining days were obviously bad.

Supporting Graphs :

Ideal Day :



Step 3:

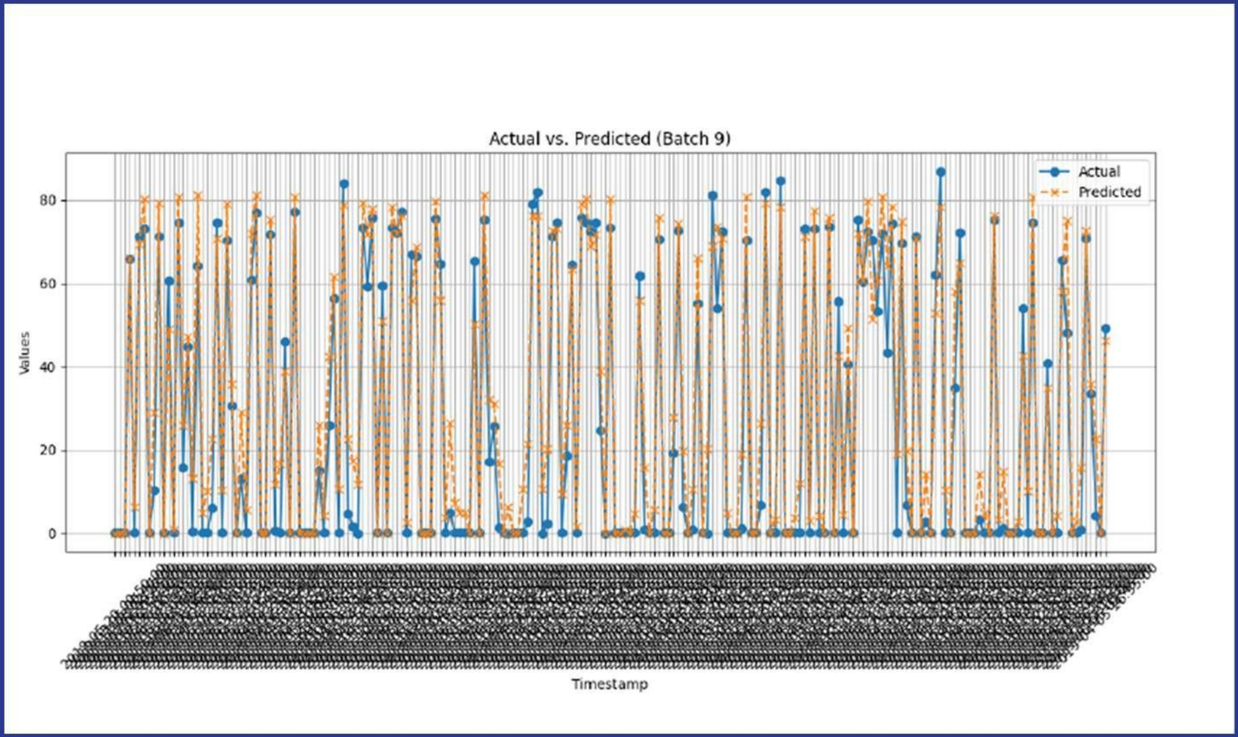
Converting Good Days into Very Good Days

Now for all the timestamps within a good days , we detected the outliers by measuring the squared error from their expected value(the value corresponding to that respective timestamp taken from the ideal day dataset).

For those timestamps which had an error greater than a certain threshold , we labelled them as outliers and replaced those values by the value corresponding to that respective timestamp taken from the ideal day dataset.

We observed that this improved the quality of the good data set by a significant amount.

Supporting Graphs :



Step 4:

Creating new Features using Feature Creation

As we had not been given any features to begin with we had to engineer all the features from scratch.

After simply observing the plots , it is evident that the current depends mainly on hours and minutes. So we initially created numerous features including their higher powers and even their mutual products of all possible combinations (around 28 new columns).

Next we even had to normalize these data columns or else the model will be unwantedly biased towards the higher value terms.

Step 5:

Applying MLR and Feature Elimination

Now we applying our ML model on 80 percent of randomly selected good data.

We check the p-values of all the feature added and then start eliminating features one by one based on their high p values.

Finally we test the ML model on the testing data and get a R2 squared values of about 0.961 and an F-statistic value of close to 10^4 thus proving that our model is very much acceptable.

Supporting Graphs :

Quality of our ML Model

Dep. Variable:	Corrected_HT R Phase Current	R-squared (uncentered):	0.961
Model:	OLS	Adj. R-squared (uncentered):	0.961
Method:	Least Squares	F-statistic:	2.293e+04
Date:	Sun, 08 Oct 2023	Prob (F-statistic):	0.00
Time:	13:24:42	Log-Likelihood:	-60362.
No. Observations:	16819	AIC:	1.208e+05
Df Residuals:	16801	BIC:	1.209e+05
Df Model:	18		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Hours2	9084.1181	301.240	30.156	0.000	8493.656	9674.580
Hours3	-1.142e+04	282.967	-40.355	0.000	-1.2e+04	-1.09e+04
Hours4	4981.4544	107.167	46.483	0.000	4771.396	5191.512
Hourscb	-1260.2391	113.042	-11.148	0.000	-1481.813	-1038.666
Hourssq	2960.9336	236.796	12.504	0.000	2496.789	3425.079
hm2	2869.7176	488.529	5.874	0.000	1912.149	3827.286
h2m2	-3855.6196	591.290	-6.521	0.000	-5014.609	-2696.630
h3m2	3613.5338	555.307	6.507	0.000	2525.074	4701.994
h4m2	-1300.5187	210.259	-6.185	0.000	-1712.649	-888.388
hcbm2	1117.1161	222.869	5.012	0.000	680.269	1553.963
hsqm2	-2444.2112	466.350	-5.241	0.000	-3358.307	-1530.116
hmsq	2383.3237	562.676	4.236	0.000	1280.419	3486.229
h2msq	-2887.7794	681.766	-4.236	0.000	-4224.112	-1551.447
h3msq	2470.2362	640.419	3.857	0.000	1214.948	3725.524
h4msq	-813.5080	242.528	-3.354	0.001	-1288.889	-338.127
hcbmsq	1004.3361	256.070	3.922	0.000	502.411	1506.261
hsqmsq	-2155.9321	536.273	-4.020	0.000	-3207.083	-1104.781
Hoursn	-4337.3490	248.568	-17.449	0.000	-4824.569	-3850.129

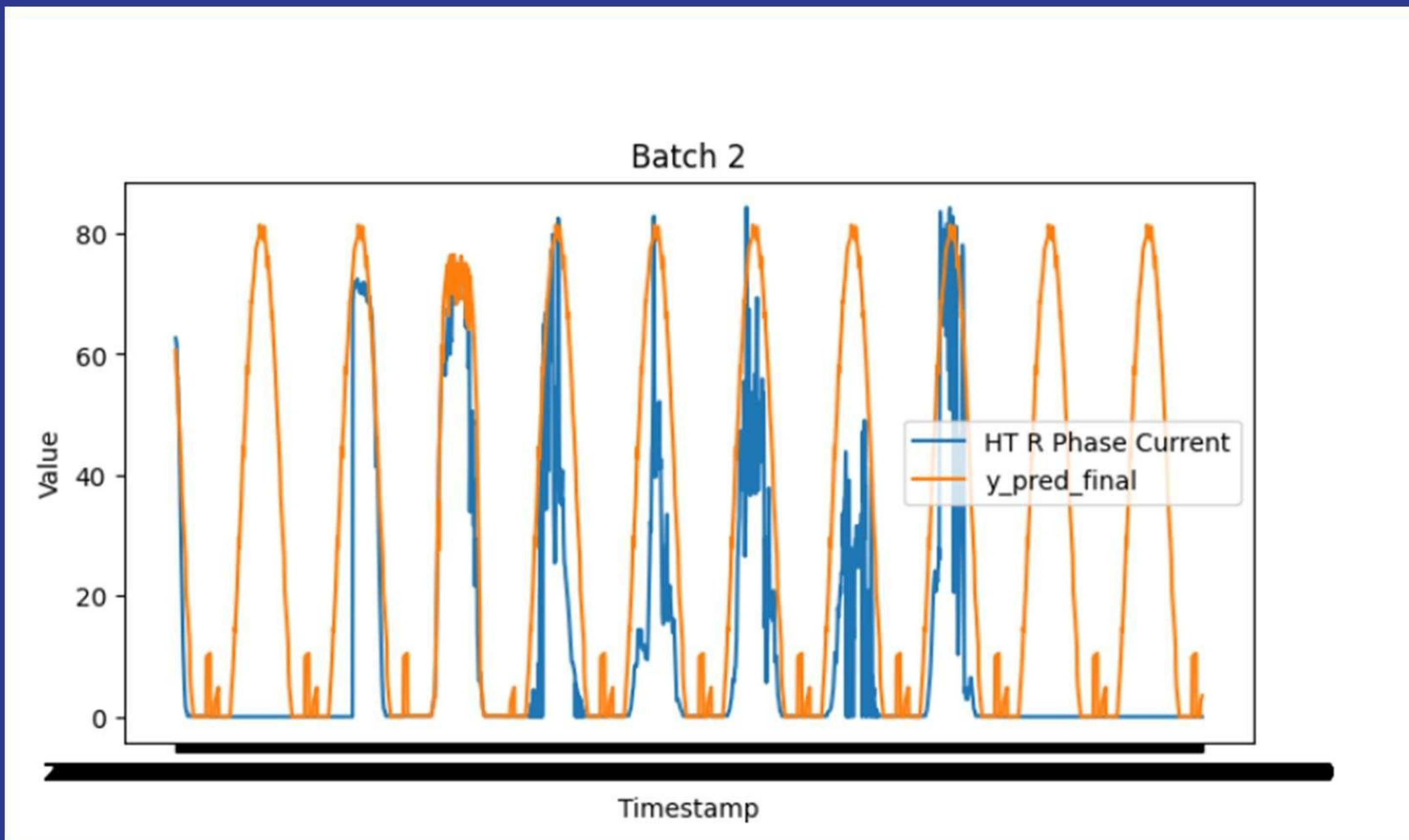
Step 6:

Applying the ML model to predict the bad and missing days

On applying the ML model to the bad and missing data and then plotting the days after prediction we see that our model is predicting very accurately .

Also we see that the errors for the bad days have been minimised.

Supporting Graphs :



Descriptive Stats

Good Days (Initial)

```
Mean: 28.11250945790874
Median: 0.0
Mode: 0.0
Range: 197.63978102711116
Variance: 3329.907659999717
Standard Deviation: 57.705352091463034
Skewness: 1.7793933653971596
Kurtosis: 1.5778208096222714
25th Percentile: 0.0
50th Percentile (Median): 0.0
75th Percentile: 0.0
```

Good Days (Final)

```
Mean: 9.219760851153858
Median: 0.0
Mode: 0.0
Range: 61.894958240195585
Variance: 334.87626601667694
Standard Deviation: 18.29962475070669
Skewness: 1.6121452709943187
Kurtosis: 0.8926904616591456
25th Percentile: 0.0
50th Percentile (Median): 0.0
75th Percentile: 0.0
```

Descriptive Stats (Contd)

Bad Days (Initial)   Bad Days (Final)

```
Mean: 288.555738289845
Median: 283.82094442898966
Mode: 0.0
Range: 760.7417355522905
Variance: 75692.91805862868
Standard Deviation: 275.1234596660719
Skewness: 0.3493195739738685
Kurtosis: -1.238433710267386
25th Percentile: 0.0
50th Percentile (Median): 283.82094442898966
75th Percentile: 507.51842406609006
```

```
Mean: 80.76414078482364
Median: 134.66079009759872
Mode: 134.66079009759872
Range: 134.66079009759892
Variance: 4358.433425662783
Standard Deviation: 66.01843246899143
Skewness: -0.4081528644961082
Kurtosis: -1.8432466351564454
25th Percentile: 0.0
50th Percentile (Median): 134.66079009759872
75th Percentile: 134.66079009759872
```

Descriptive Stats (Contd)

Entire Data (Initial) Entire Data (Final)

```
aily_errors_old.describe()
```

	Date	Root_Sum_Squared_Error
count	352	352.000000
mean	2019-06-16 12:00:00	315.949462
min	2018-12-23 00:00:00	0.000000
25%	2019-03-20 18:00:00	104.736616
50%	2019-06-16 12:00:00	283.820944
75%	2019-09-12 06:00:00	507.518424
max	2019-12-09 00:00:00	760.741736
std	NaN	251.202940

```
daily_errors.describe()
```

	Date	Root_Sum_Squared_Error
count	352	352.000000
mean	2019-06-16 12:00:00	89.748169
min	2018-12-23 00:00:00	0.000000
25%	2019-03-20 18:00:00	38.776326
50%	2019-06-16 12:00:00	134.660790
75%	2019-09-12 06:00:00	134.660790
max	2019-12-09 00:00:00	134.660790
std	NaN	56.846618

One alternate approach !

One alternate approach could have been used while identifying the bad days would have been to set an appropriate threshold of maximum and minimum acceptable values of htr current and check how many times in a particular day the current value crossed that threshold and use it to classify the days.

Also we could have used an alternate Machine Learning Approach called Support Vector Machines which would have eliminated the need to engineer the new features and would also create a non-linear decision boundary similar to our current challenge.

Can this improved data set be really used for creating model :

Yes , according to us this improved data can indeed be used to create the model because it captures the true nature of the Htr current data and then it uses this knowledge of the true nature to predict the bad and missing days.

Slight errors which might have crept in due to human negligence or other anomalies are removed to give a cleaned data to work on.

THANK YOU !!