DS-203 : Exercise -7

Team Members
Saarthak Krishan 22B3959
Sanat Agrawal 22B3919
Yash Gupta 22B1813

The problem

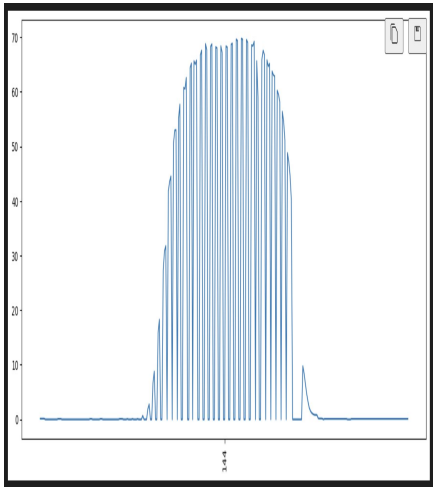
Context

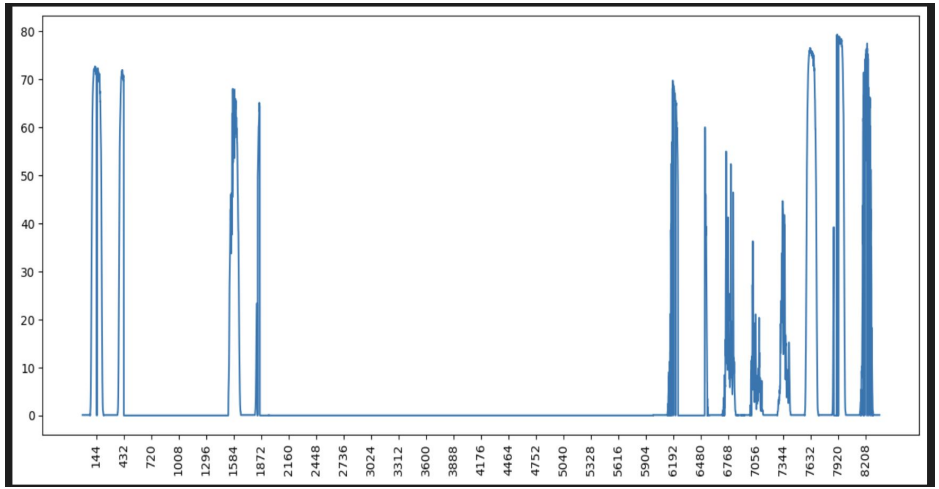
- Transformer current data, sampled every 5 minutes for ~280 days
- Data source: a solar power generation site
- Data was choppy (bad) due to various reasons on some days
- Data could not be sampled due to various reasons on ~80 days
- Good data also slightly noisy readings, but seemingly easy to fix.

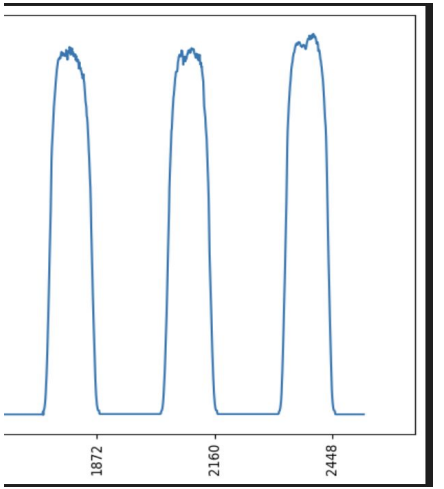
Problem statement

Can the data be “de-noised”? That is, can the data set’s quality be improved with the help of the data itself?

Visualization







Bad Data

Missing Data

Good Data

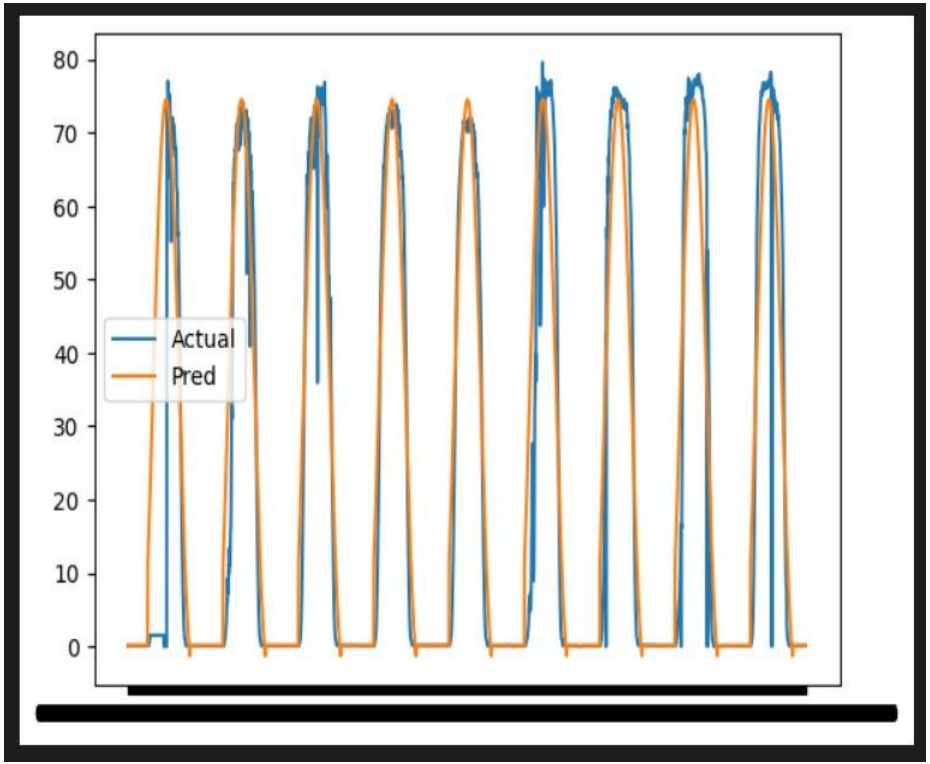
Solution:

Creating an ML model to Denoise the data!

The following Steps were followed to successfully denoise the data:

- Classification of data into good, bad and missing
- Improvement of good data to make it even better to analyse
- Training and testing the model with the good (better) dataset.
- Implementing the model to bad and missing days to get a predicted dataset of all 285 days.

Achievements:



OLS Regression Results

Dep. Variable:	HT R Phase Current	R-squared:	0.852
Model:	OLS	Adj. R-squared:	0.852
Method:	Least Squares	F-statistic:	2.648e+04
Date:	Thu, 12 Oct 2023	Prob (F-statistic):	0.00
Time:	23:28:19	Log-Likelihood:	-89654.
No. Observations:	23040	AIC:	1.793e+05
Df Residuals:	23034	BIC:	1.794e+05
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.0871	0.002	39.742	0.000	0.083	0.091
Time	1.0396	0.026	39.715	0.000	0.988	1.091
T2	-0.0675	0.001	-63.659	0.000	-0.070	-0.065
T4	-9.748e-06	1.05e-07	-92.478	0.000	-9.95e-06	-9.54e-06
T3	0.0013	1.56e-05	84.067	0.000	0.001	0.001
T5	3.092e-08	3.32e-10	93.189	0.000	3.03e-08	3.16e-08
T6	-3.548e-11	3.94e-13	-89.952	0.000	-3.63e-11	-3.47e-11

Omnibus:	8515.040	Durbin-Watson:	0.087
Prob(Omnibus):	0.000	Jarque-Bera (JB):	63963.611
Skew:	-1.584	Prob(JB):	0.00

Implementation

Classification of the Data:

1. Clipping of the data: Since the ‘Current’ is zero or at very low value everyday before and after certain time. We considered only the time interval when the transformer is actually working.
2. Missing Days: Now we considered the value of current during the clipped interval and put up a condition: If the current is equal to 0 at more than 30 Timestamps, then it is considered missing.
3. Good Days: After classifying the missing days, on the remaining days we put up another condition to find the good days:
    a. We found out the local standard deviation i.e. for every data we found the standard deviation of it and 4 points around it.
    b. If the standard deviation is more than 10 and if it has happened more than 10 times then the day will be classified as bad
    c. Thus, the remaining data that we have is of good days.

Making Good data “Better”:

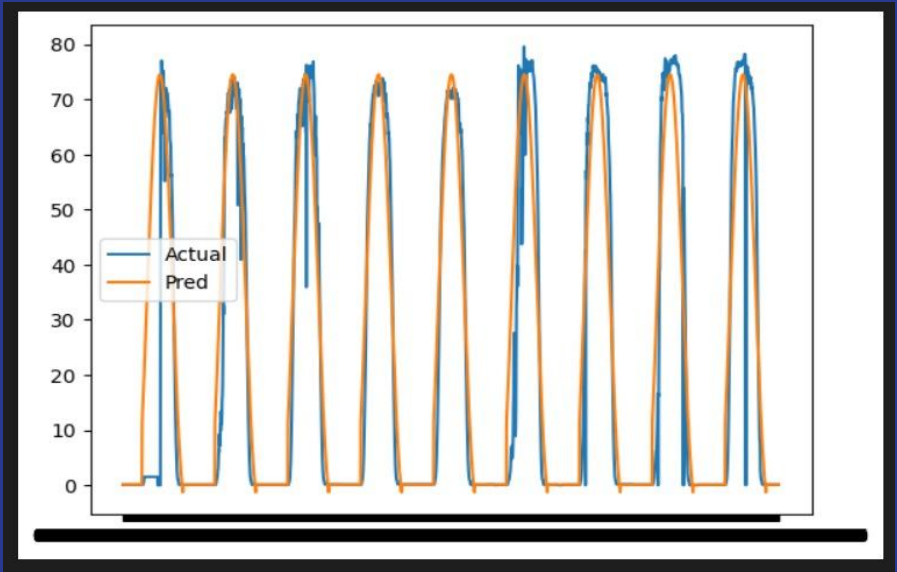
1. Since the data is still noisy even though it is good, we can improve its quality.
2. The noise can be visualised as sudden dip in the value of the current.
3. To improve the quality, we compared the current at a particular timestamp with the value of current at the previous timestamp.
4. Now if its value is less than a particular threshold (taken to be 30) then we change it by the local mean i.e. the mean of the previous value and 4 values around it.
5. Now the sudden dips in the value are corrected and data is free of these noises.

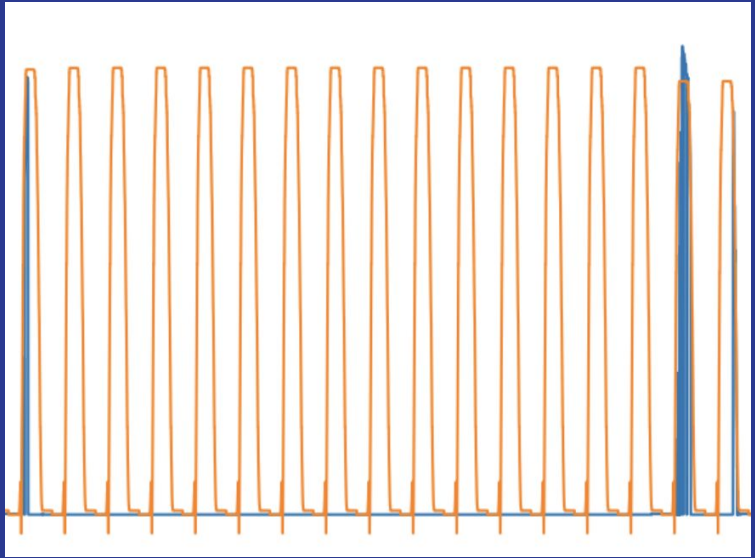
Making the ML Model:

1. To train the ML model, we first selected 80% of the good data.
2. Then we applied Multiple Linear regression on all the days by taking the ‘HT R Phase Current’ as the dependent variable and ‘Time’, ‘Time^2’, ‘Time^3’, etc. and a constant.
3. Then we procured the values of their coefficient.
4. The coefficient found above was then used to predict the test data (remaining 20%). Both the datas was then compared by finding the accuracy
5. Then after dropping and adding other independent variables (Feature Engineering) the model was improved until the accuracy was improved.

Denoising the Data:

1. Now that everything is ready, we can use it to predict the current of all the 285 days.
2. We can plot and see that the ML model has worked quite well. The bad data is now smooth and also the values for the missing days is now predicted quite nicely.





Why our approach is the best one!

OLS Regression Results

Dep. Variable:	HT R Phase Current	R-squared:	0.852
Model:	OLS	Adj. R-squared:	0.852
Method:	Least Squares	F-statistic:	2.648e+04
Date:	Thu, 12 Oct 2023	Prob (F-statistic):	0.00
Time:	23:28:19	Log-Likelihood:	-89654.
No. Observations:	23040	AIC:	1.793e+05
Df Residuals:	23034	BIC:	1.794e+05
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.0871	0.002	39.742	0.000	0.083	0.091
Time	1.0396	0.026	39.715	0.000	0.988	1.091
T2	-0.0675	0.001	-63.659	0.000	-0.070	-0.065
T4	-9.748e-06	1.05e-07	-92.478	0.000	-9.95e-06	-9.54e-06
T3	0.0013	1.56e-05	84.067	0.000	0.001	0.001
T5	3.092e-08	3.32e-10	93.189	0.000	3.03e-08	3.16e-08
T6	-3.548e-11	3.94e-13	-89.952	0.000	-3.63e-11	-3.47e-11

Omnibus:	8515.040	Durbin-Watson:	0.087
Prob(Omnibus):	0.000	Jarque-Bera (JB):	63963.611
Skew:	-1.584	Prob(JB):	0.00

- The technique of finding the local mean and variance provides our solution an edge over the others.
- The clipping of the data to focus only on the interval when the transformer is actually working.
- All this is justified by the value of accuracy and R^2.

Alternate Solution

- There are many other methods to train and test the data.
- One possible method is to apply decision tree algorithm available in python in xgboost library.
- It minimizes the loss function by iteratively adding decision trees while emphasizing the correct prediction of previously misclassified data points.
- But since this method lacks transparency we dropped this approach.

Challenges deep-dive

Challenge 1

Transformer wasn't ON throughout the day

This was tackled by clipping of the data.

Challenge 2

Improving good data

It was quite challenging to precisely point out the noise and then replacing it by some appropriate values.

Challenge 3

Finding appropriate variables

Since the curve can't be fitted by some simple polynomials of degree 1 or 2. The best assumption was approximating the function as a gaussian function with some variables with odd powers of time.