Reference Energy Disaggregation Dataset

Primary Topic: Time Series

Course: 2020-201400174-2A – Group: 72 – Submission Date: 2020-04-18

Deepika Jangamguravepalli Bramhanandareddy
University of Twente
d.jangamguravepallibramhanandareddy@student.utwente.nl

Sathvik Guru Rao
University of Twente
s.gururao@student.utwente.nl

ABSTRACT

In this paper, time series forecasting is performed on the Reference Energy Disaggregation Dataset. The appliance with most energy consumption in all the houses- lighting is chosen as the topic of the paper. A preliminary analysis of the data was done to check for the stationarity of the time series data. A comparison of ARIMA and Multi Layer Perceptron algorithm for forecasting is performed using the mean squared error as the evaluation metric. The MLP model has performed well in most of the houses but the ARIMA model also has good prediction.

KEYWORDS

Energy disaggregation, dickey fuller test, ARIMA, MLP

1 INTRODUCTION

Among 40% of the co2 emission caused by electricity generation which contribute almost half to global warming for a sustainable future we need to be aware of the power consumption in large to small sectors and developing an efficient energy consumption is essential at this current time and one of the common goal of Energy Disaggregation is to make consumers more aware of their power consumption.

Energy Disaggregation is a method to estimate the electricity consumption of home appliances, where disaggregation allows us to take the whole building load, and separate into each appliance specific data. Energy disaggregation is one of the most anticipated analytical technologies applied to home electric load data to understand the utilization.

The purpose of this paper is to determine which algorithm can give better forecast and also, if selected data is having any seasonality which might reflect in future predictions.This study has two parts,

- Analysis on data for season trends to check the data inconsistency in selected appliance for the research
- Comparison of Time series and machine learning algorithm for forecasting the future data.

Mainly our study focuses on the prediction of lighting appliances as it is one of the most common appliances in all the houses and also it is a major appliance where energy consumption is higher after refrigerator. After selecting the appliance, on appliance data we performed a dickey fuller test to check for the stationary of the data as it is important for the time series data to be stationary. Furthermore, ARIMA model and Multilayer perceptron model was used to forecast and the results were compared to get the best model. The models were evaluated using mean squared error metric.

2 BACKGROUND

In this section, a few important topics which are related to the project are described briefly, starting with the dataset, and the algorithms and methods which can be helpful in understanding the paper further.

2.1 Dataset

The dataset used in this paper is lowfreq.tar.bz2 from the Reference Energy Disaggregation Data Set(REDD) by Kolter and johnson [5]. The dataset consists of 6 different house energy consumption data with each house having different appliances, data collection is done with 3-second frequency. The table below shows the number of appliances in each house and the number of days the data is collected.

House	Appliances	Days	continuous days
1	20	23	11
2	11	16	15
3	22	26	13
4	20	26	16
5	26	9	7
6	17	17	8

Table 1: Dataset Description

The collection of data is not continuous, if observed, the collected data of house 1 consists is 23 days, but only 11 days of data is continuous. In time series analysis the data should be continuous for the forecast.

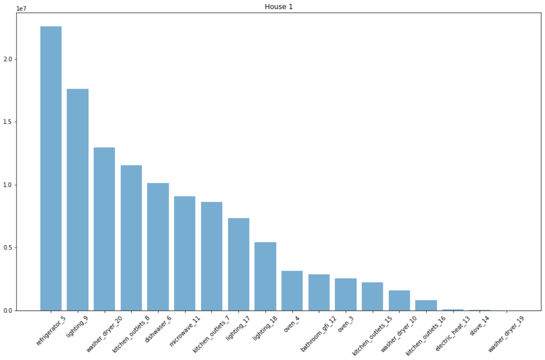


Figure 1: Total Energy Consumption

For selecting the disaggregation appliance, the total energy consumption per appliance was considered, to identify the highest energy consumption appliance. For instance, among all houses, lighting seems to be using a lot of power.

2.2 Data Pre-Processing

In data processing, the lightning in each house is having multi-column information for instance lighting in the kitchen, living room, etc each column information is aggregated into a single column as Lightning for uniformity in model training. Without it, we need to train each lighting area of each house which is not an ideal way. After aggregation each house data dimensions are as follows.

House	Before Aggregation	After Aggregation
1	(406748, 25)	(406748, 11)
2	(316840, 13)	(316840, 9)
3	(376150, 27)	(376150, 13)
4	(428076, 25)	(428076, 12)
5	(77451, 33)	(77451, 14)
6	(192192, 21)	(192192, 12)

Table 2: Aggregation of Dataset

2.3 Augmented Dickey Fuller Test

In time series forecast, testing the stationarity is important [4] step. The Augmented dickey fuller(ADF) [6] test is a statistical significance test, with hypothesis testing to accept or reject the hypothesis that the data is stationary. In the statistical test the p-value reported decides whether data is stationary or not. ADF test is a category of "Unit Root Test" which is the proper method to test the stationarity of time series.

In time series, data will be non stationary when the unit root value of alpha is 1, the presents of unit root value indicates the data is non stationary, before getting into ADF, first we need to understand Dickey fuller test(DFT), it is also a unit root test that test the null hypothesis that alpha = 1 and and DFT can be formulated by following equation, where alpha is the coefficient of the first lag on Y.

Null Hypothesis (H0) : alpha =1

$$y_t = c + \beta t + \alpha y_{t-1} + \Phi \Delta Y_{t-1} + e_t \tag{1}$$

where, Yt - time series at time 't', y(t-1) = lag 1 of time series, and delta Y(t-1) = first difference of the series at time (t-1)

ADF test is based on above equation which expands on the dickey fuller test equation which includes higher order regressive process, and ADF is formulated with the following equation, but the key point to remember is here as alpha value is 1, so the p value obtained should be less then the significance level of <0.05 in order to reject the null hypothesis which can be test by using the *adfuller()* from statsmodels library package which is used in this paper.

adfuller() accepts the number of lags, by default the value of lag will be 12*(nobs/100)1̂/4 ,but in this paper algorithm computed the optimal iteration number

2.4 ARIMA model

AutoRegressive Integrated Moving Average model[2], The most commonly used model for forecasting a time series, also it can be viewed as a filter that tries to separate the single form the noise, and then generalize it for the future to obtain forecasts. ARIMA is the combination of autoregression and moving average model, The model takes three parameters p,d,q.

- p - order of autoregression.
- d - degree of differencing.
- q - order of moving average.

The ARIMA forecasting equation for a stationary time series is a linear equation which can be constructed as follows, where Here the moving average parameters (θ) 's are defined as a negative sing in the equation

$$Y^t = \mu + \Phi_1 Y_{t-1} + \cdots + \Phi_p y_{t-p} - \Theta_1 e_{t-1} - \cdots - \Theta_q e_{t-q} \tag{2}$$

The autoregression part of the model uses the relationship between current observation and previous observations.The integration part of the model uses differencing of the observations to transform the time series data to stationary. The moving average part of the model uses the pattern in the data and the residual error obtained from moving average.The arima model prepares the data to be stationary by using the integration. This removes the trend or seasonality in the time series data if it exists.

2.5 Multi Layer Perceptron

The Multi Layer Perceptron(MLP) is a feed forward neural network approach to time series forecasting[1]. As only the lighting data from each house is considered, the univariate multilayer perceptron is considered which is used for a single series of observations. This model learns from a certain number of past observations to predict the next observation. Multilayer perceptron is a fully connected sequential model with input layer, hidden layer and an output layer.The input layer takes the number of past observations considered as the input dimension. The hidden layer and output layer has activation function and loss function.

2.6 Mean Squared Error

The Mean Squared Error(MSE) is used as an evaluation metric for the forecasting models. The MSE calculates the squared difference between the predicted values and the test values. The value is then averaged. In this paper, the mean squared error is calculated using the mean_squared_error function from *sklearn.metrics* package. The function takes the test values and the predicted values as parameters.

$$MSE = \frac{1}{n} \sum_{t=1}^{n} (test\_value - predict\_value)^2 \tag{3}$$

3 APPROACH

The lighting data of each house is considered for the algorithms. The data was re-sampled into hourly data using the pandas *resample()* function with 'H' parameter which aggregates the data by hour. The figure 2 and figure 3 shows the diffrence of resampling. Resampling has removed the anomalies. This is done to reduce the computation time and make it more efficient. The data was then split into 80 percent for training and 20 percent for testing.

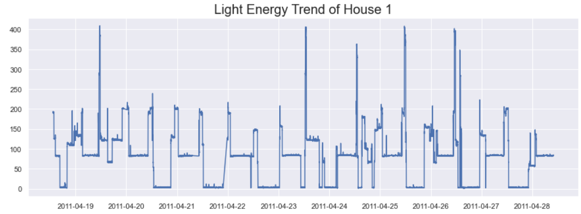


Figure 2: house 1 data without resampling

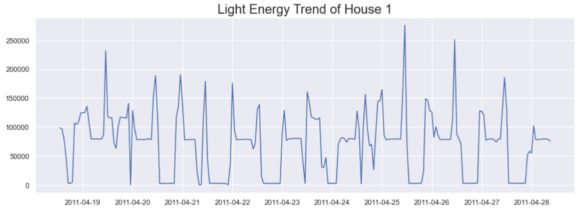
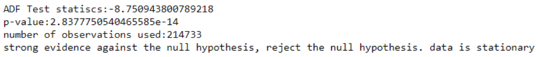


Figure 3: House 1 data after resampling into hourly.

] A function was created to check for the stationarity of the data using the adfuller() function from statsmodel library. The function performs a statiscal test on the given data. The p-value of the test is considered to decide whether the given time series data is stationary or not.

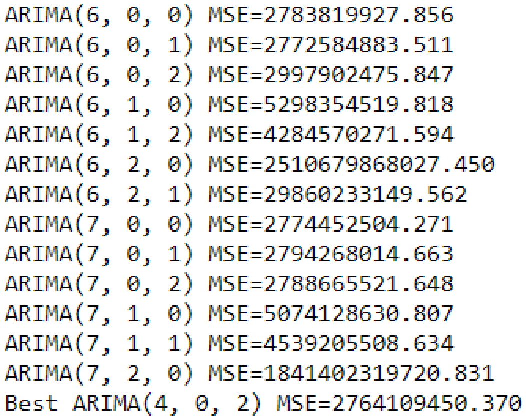


```
ADF Test statiscs:-8.750943800789218
p-value:2.8377750540465585e-14
number of observations used:214733
strong evidence against the null hypothesis, reject the null hypothesis. data is stationary
```

Figure 4: Augmented Dickey Fuller test on House 1 data

From the above figure 4, the adfuller test gives the statiscal test results. The p-value is less than 0.05, so the null hypothesis is rejected. This means that the data does not need any differencing to transform into stationary data. The grid search [3] for searching the parameters has also given the result as 0 for d value which confirms the stationarity. The ARIMA model was created using the *statsmodels* library package. The model takes three parameters p,d,q which was selected using grid search. The value of P was between the range of 0 to 7 , and the value of d and q were between the range of 0 to 3. The arima model was created for each combination of the parameters and evaluated with the predicted and tested values. The parameters with the least MSE value is chosen and the next visualization of the results for that model is done. Some of the houses data was stationary, so the value of *d* was zero which was also found by the grid search.This is shown in figure 5 The arima model was fitted with training data and tested on the test data.

The prediction was calculated by iterating over the number of test data. For each iteration, the arima model was created and the predicted value was stored in a list to use it for the evaluation. In this method the error rate was very high. To reduce the error, after each iteration, the test data for that iteration was append to the training data and in the next iteration that data was also used for training. In this way the training data was increased as the number of iterations increased. This method has reduced the error rate.

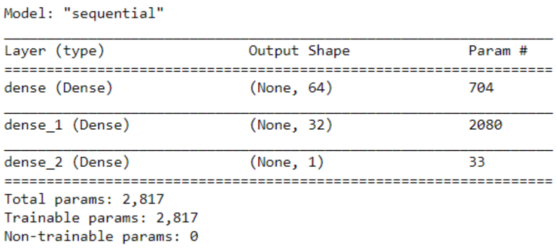


```
ARIMA(6, 0, 0) MSE=2783819927.856
ARIMA(6, 0, 1) MSE=2772584883.511
ARIMA(6, 0, 2) MSE=2997902475.847
ARIMA(6, 1, 0) MSE=5298354519.818
ARIMA(6, 1, 2) MSE=4284570271.594
ARIMA(6, 2, 0) MSE=2510679868027.450
ARIMA(6, 2, 1) MSE=29860233149.562
ARIMA(7, 0, 0) MSE=2774452504.271
ARIMA(7, 0, 1) MSE=2794268014.663
ARIMA(7, 0, 2) MSE=2788665521.648
ARIMA(7, 1, 0) MSE=5074128630.807
ARIMA(7, 1, 1) MSE=4539205508.634
ARIMA(7, 2, 0) MSE=1841402319720.831
Best ARIMA(4, 0, 2) MSE=2764109450.370
```

Figure 5: Grid Search for ARIMA model parameters.

The multilayer perceptron model takes a sequence of data as input and predicts the next value of the data. The model learns the pattern from the sequence and predicts according. The lighting data was divided into sequences of data. For this model, 10 past observations were considered and the 11th observation is to be predicted and used for testing. A function was created to create the sequence. The training data was first transformed into sequences of data and used to train the model. The test data was also converted into sequences of data and used for testing. The fitted model is used to predict the values for the next time steps using the predict() function.

This value is evaluated against the test values and the error was calculated. The model has four layers , one input layer, two hidden layers and one output layer. The input layer takes the number of previous data considered , in this case 10 as the input dimension. The model is fully connected by using the dense layer of keras library. The *relu()* function is used as the activation function for the layers.After the model is built, it was compiled using the keras *compile()* function with mean squared error as the loss function and '*adam*' optimizer is used. The model was trained for 200 epochs with mean squared error loss function. The loss function reduces the error after each epoch.



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 64)                704
_________________________________________________________________
dense_1 (Dense)              (None, 32)                2080
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 33
=================================================================
Total params: 2,817
Trainable params: 2,817
Non-trainable params: 0
```

Figure 6: Model summary

4 EVALUATION AND RESULTS

In this section the results of the models are discussed. The visualization of results of house 1 data is discussed here. The same method is used for other houses as well, but the visualization are mentioned in the section **??**. The forecasting models were evaluating using mean squared error. The error for each house and for each algorithm is given in table 3.

House	ARIMA	MLP
1	1754298965.57	1397073792.00
2	693243238.158	1633513728.00
3	9520231635.205	21007241216.00
4	1156238019.987	2667508224.00
5	1699436627.791	2060196224.00
6	20673871.367	15491669.00

Table 3: Mean Squared Error of ARIMA and MLP model

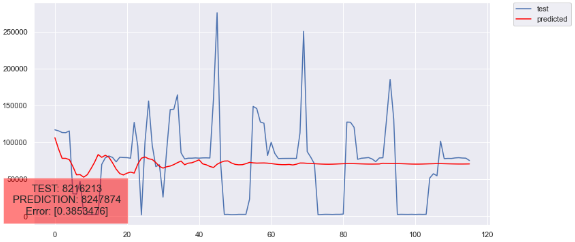


Figure 7: ARIMA model - fig 1.

The house data was tested on both the models. The ARIMA model model when tested with just using the training data and not iteratively adding the test data for further prediction has the result as shown in figure 7.The ARIMA model implemented by using each test data after the iteration has given a better result and the error has decreased extremely. This method is used on all the house data for prediction.

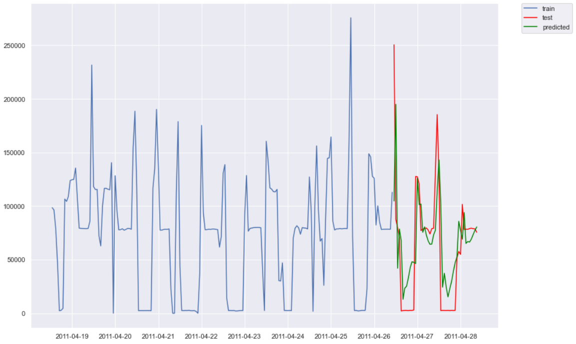


Figure 8: ARIMA model - fig 2.

The figure 9 shows the result of Mutlilayer Perceptron model. The model was first evaluated with the training data which is the learning part. The blue line in the plot represents the true value of house 1 lighting data. The predicted values for training data is represented in orange line. The model fitted on the training data is then used on the test data. The predicted values for test data is represented by the green line.

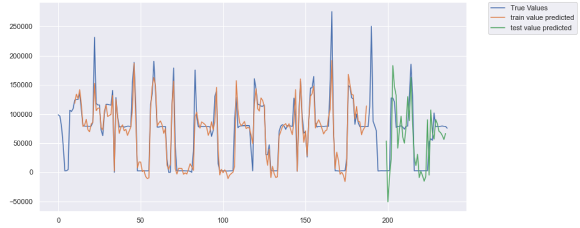


Figure 9: MLP forecast

4.1 Best forecasting of models

In this section, the models which have performed well for each house is plotted.

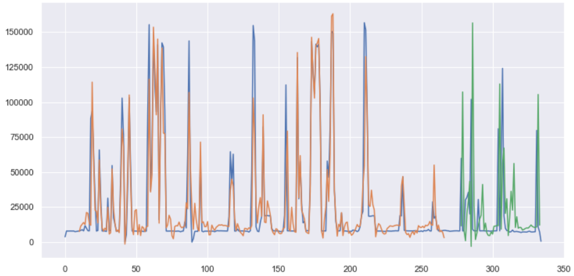


Figure 10: MLP model forecast for house 2

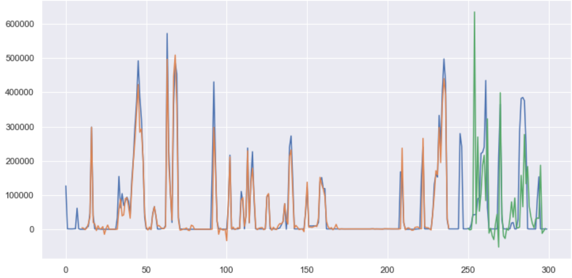


Figure 11: MLP model forecast for house 3

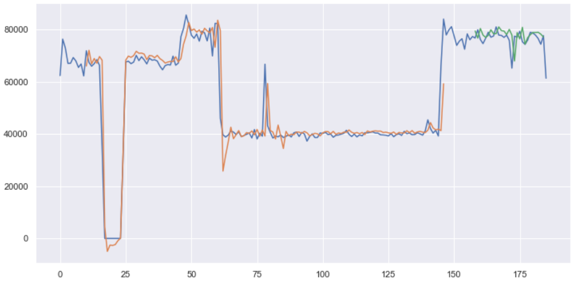


Figure 12: MLP model forecast for house 6

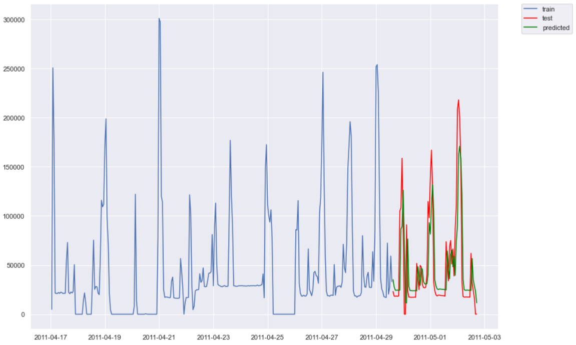


Figure 13: ARIMA model forecast for house 4

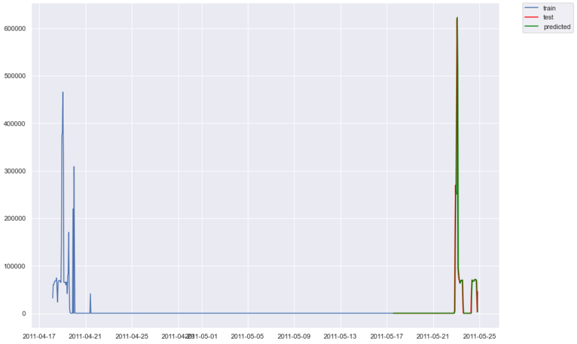


Figure 14: ARIMA model forecast for house 5

5 DISCUSSION

The ARIMA model and MultiLayer Perceptron model has given some good results. Comparitively, the MLP model has less error in most of the houses. choosing the parameters for ARIMA models is a time consuming problem. The grid search gives the best parameters which has less error but the time taken to compute the result for each combination of results takes long time. The MLP models are efficient but the drawback which can be observed from the results is that, MLP models has predicted negative values of power consumption which is impossible. This has to be considered when using the model for future forecasting.

6 CONCLUSIONS

We can conclude that the multilayer perceptron model has the best forecast compared to ARIMA model and with further hyper parameter optimizations like using more hidden layers and different activation functions, the model can perform better in all the houses and can be used to forecast for the next time periods. In future work the whole dataset can be used and the missing data in between days can be duplicated from previous data and a study can be done on how the duplicate data affect/help in the forecasting.

REFERENCES

[1] [n. d.]. Crash Course On Multi-Layer Perceptron Neural Networks. https://machinelearningmastery.com/neural-networks-crash-course/. Accessed: 2020-08-15.

[2] [n. d.]. How to Create an ARIMA Model for Time Series Forecasting in Python. https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/. Accessed: 2020-12-10.

[3] [n. d.]. Hyperparameter Optimization With Random Search and Grid Search. https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search/. Accessed: 2020-09-04.

[4] Eduard Baumöhl and Štefan Lyócsa. 2009. *Stationarity of time series and the problem of spurious regression.* MPRA Paper 27926. University Library of Munich, Germany. https://ideas.repec.org/p/pra/mprapa/27926.html

[5] J Kolter and Matthew Johnson. 2011. REDD: A Public Data Set for Energy Disaggregation Research. *Artif. Intell.* 25 (01 2011).

[6] Antonio Montañés bernal and Andreu Sanso. 2001. The Dickey-Fuller Test Family and Changes in the Seasonal Pattern. *Annales d'Economie et de Statistique* 61 (01 2001), 73–90. https://doi.org/10.2307/20076270

A APPENDIX

A.1 ARIMA model forecast results

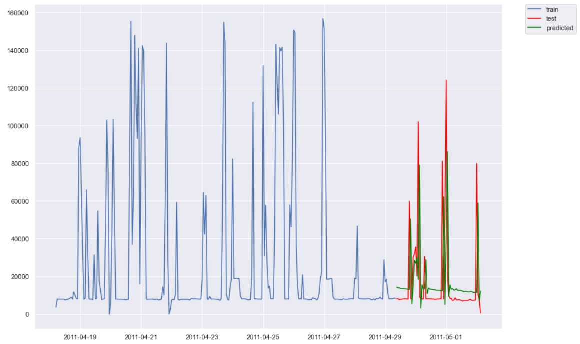


Figure 15: ARIMA model forecast for house 2

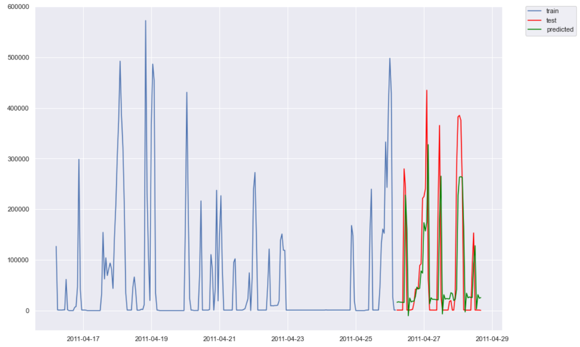


Figure 16: ARIMA model forecast for house 3

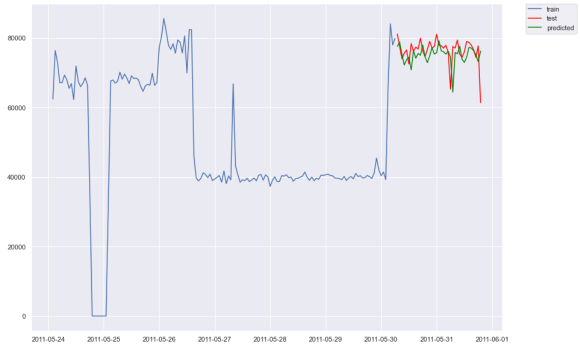


Figure 17: ARIMA model forecast for house 6

A.2 MLP model forecast results

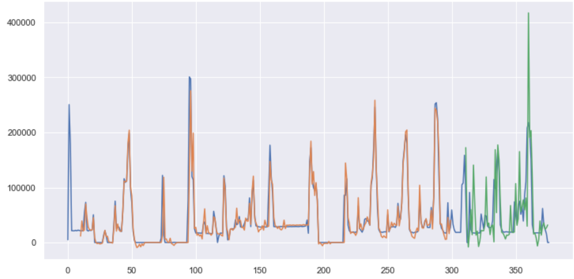


Figure 18: MLP model forecast for house 4

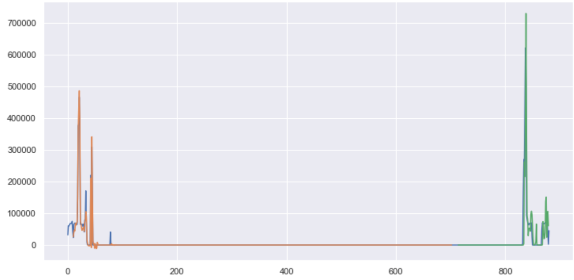


Figure 19: MLP model forecast for house 5