Contents

List of figures:

1 Scientific Method

This chapter outlines the scientific methods used to collect, describe, and analyse the data.

1.1 Data collection and description

The thesis conducted in this work use a real data set containing minutely electricity consumption for the period between December 2006 and November 2010 (47 months). The data set consisted of 2075259 readings from an individual household located in Sceaux, France (*1*)

date	Date in format dd/mm/yyyy
time	time in format hh:mm:ss
global_active_power: household global minute-averaged active power	in kilowatt
global_reactive_power: household global minute-averaged reactive power	in kilowatt
voltage: minute-averaged voltage	in volt
global_intensity: household global minute-averaged current intensity	in ampere
sub_metering_1: energy sub-metering No. 1 , It corresponds to the kitchen, containing mainly a dishwasher, an oven and a microwave (hot plates are not electric but gas powered)	in watt-hour of active energy
sub_metering_2: energy sub-metering No. 2, It corresponds to the laundry room, containing a washing-machine,a tumble-drier, a refrigerator and a light.	in watt-hour of active energy
sub_metering_3: energy sub-metering No. 3, It corresponds to an electric water-heater and an air-conditioner.	in watt-hour of active energy

Table 1 Variables available in the data set

1.1.1 Definitions of data set variables

Active power: The portion of power that, averaged over a full cycle of the AC waveform, lands up in net transfer of energy in one direction is believed as active power (more commonly called real power to avoid ambiguity especially in discussions of loads with non-sinusoidal currents) (*2*).

Reactive power: The portion of power due to stored energy, which returns to the source in each cycle, is known as instantaneous reactive power, and its amplitude is that the measure of reactive power.

Voltage: Voltage is that the pressure from an electrical circuit's power source that pushes charged

electrons (current) through a conducting loop, enabling them to undertake to to figure like illuminating a light-weight.

Intensity: Intensity is that the speed, per unit time, at which current is transferred by an circuit.

1.2 Data Processing and Data cleaning

Among all the steps involved in data analysis, data preparation, though seemingly less problematic, is in fact one that requires more resources and more time to be completed. The collected data are often collected from different data sources, each of which will have the data in it with a different representation and format. So, all of these data will have to be prepared for the process of data analysis.

The preparation of the data is concerned with obtaining, cleaning, normalizing, and transforming data into an optimized data set, that is, in a prepared format, normally tabular, suitable for the methods of analysis that have been scheduled during the design phase. Many are the problems that must be avoided, such as invalid, ambiguous, or missing values, replicated fields, or out-of-range data. The dataset used in thesis contains some missing values in the measurements (nearly 1,25% of the rows). All calendar timestamps are present in the dataset but for some timestamps, the measurement values are missing: a missing value is represented by the absence of value between two consecutive semi-colon attribute separators. For instance, the dataset shows missing values on April 28, 2007.

The null values are filled by the mean of the previous day value and iterated all over the dataset. Then it shows zero null values in the dataset. The steps used to check the missing values and to fill the missing values is shown below:

The dataset is shown in the table, as discussed in the table it has 9 attributes. The data set is uploaded in Jupyter notebook and imported data python libraries such as pandas, numpy and matplotlib. The index column of the dataset is set to $date\_time$ for further data preprocessing. In the table it can be seen that the power consumed reading is started from 2006-12-16 (end of the December 2016) and the last reading was on 2010-12-11 (till December 2010). The $head()$ function displays the first five rows of the data set. The $tail()$ function displays the last five rows of the data set.

date_time	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
2006-12-16 17:00:00	152.024	8.244	8447.18	651.6	0.0	19.0	607.0
2006-12-16 18:00:00	217.932	4.802	14074.81	936.0	0.0	403.0	1012.0
2006-12-16 19:00:00	204.014	5.114	13993.95	870.2	0.0	86.0	1001.0
2006-12-16 20:00:00	196.114	4.506	14044.29	835.0	0.0	0.0	1007.0
2006-12-16 21:00:00	183.388	4.600	14229.52	782.8	0.0	25.0	1033.0

date_time	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
2010-12-11 19:00:00	143.518	6.828	13931.03	620.2	21.0	0.0	788.0
2010-12-11 20:00:00	105.200	5.090	14040.45	450.0	483.0	66.0	604.0
2010-12-11 21:00:00	66.894	5.148	14192.62	284.8	513.0	27.0	0.0
2010-12-11 22:00:00	19.232	4.574	14408.44	82.4	0.0	0.0	0.0
2010-12-11 23:00:00	38.392	2.986	14602.29	156.4	0.0	0.0	0.0

To code used check the missing values present in the data set and the results shows that it has got around 25979 missing values.

```
## Summing the nan values , it shows 25979 as missing values

np.isnan(data).sum()
```

```
Global_active_power      25979
Global_reactive_power    25979
Voltage                  25979
Global_intensity         25979
Sub_metering_1           25979
Sub_metering_2           25979
Sub_metering_3           25979
dtype: int64
```

To fill the missing values, defined a function shown below:

```
## Filling these values with previous day values using for loop.

def insert_missing(data):
    one_day= 24 * 60
    for row in range(data.shape[0]):
        for col in range (data.shape[1]):
            if np.isnan(data[row,col]):
                data[row,col] = data [row - one_day, col]
```

```
## insert missing values

insert_missing(data.values)
```

```
##checking is their any nan values , it shows zero so our dataset is cleaned now

np.isnan(data).sum()
```

```
Global_active_power      0
Global_reactive_power    0
Voltage                  0
Global_intensity         0
Sub_metering_1           0
Sub_metering_2           0
Sub_metering_3           0
dtype: int64
```

Which makes the dataset to fill the average mean of the previous day value: Then the output look like this, with no missing values. After the data preprocessing process, the quality of the data is improved without any missing values. Now, to describe the data set using depictive statistics function. Pandas *describe*() is used to view some basic statistical details like percentile, mean, standard deviation etc. of a data frame or a series of numeric values.

Table 2 Descriptive Statistics of the dataset

```
data.describe()
```

	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
count	2.075259e+06	2.075259e+06	2.075259e+06	2.075259e+06	2.075259e+06	2.075259e+06	2.075259e+06
mean	1.089418e+00	1.236871e-01	2.408364e+02	4.618401e+00	1.118474e+00	1.291131e+00	6.448635e+00
std	1.054678e+00	1.125933e-01	3.240051e+00	4.433165e+00	6.141460e+00	5.796922e+00	8.433584e+00
min	7.600000e-02	0.000000e+00	2.232000e+02	2.000000e-01	0.000000e+00	0.000000e+00	0.000000e+00
25%	3.080000e-01	4.800000e-02	2.389900e+02	1.400000e+00	0.000000e+00	0.000000e+00	0.000000e+00
50%	6.020000e-01	1.000000e-01	2.410000e+02	2.600000e+00	0.000000e+00	0.000000e+00	1.000000e+00
75%	1.526000e+00	1.940000e-01	2.428700e+02	6.400000e+00	0.000000e+00	1.000000e+00	1.700000e+01
max	1.112200e+01	1.390000e+00	2.541500e+02	4.840000e+01	8.800000e+01	8.000000e+01	3.100000e+01

The table shows the statistics values of the dataset.

Mean: It is the average of all the items in the dataset.

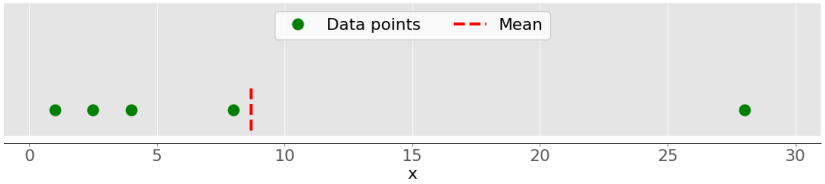


Figure 1 Mean

Median: It is the middle element in sorted dataset.

Figure 2 Median

Variance: It shows numerically how far the data points are from the mean.

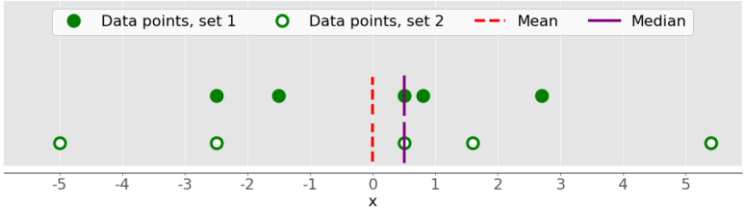


Figure 3 Variance

Standard deviation: The standard deviation is another measure of data spread. It's connected to the sample variance, as standard deviation, s, is the positive square root of the sample variance. The standard deviation is often more convenient than the variance because it has the same unit as the data points. (*3*)

Percentiles: The p percentile is the element in the dataset such that p% of the elements in the dataset are less than or equal to that value. Also, $(100 - p)$ % of the elements are greater than or equal to that value. If there are two such elements in the dataset, then the sample p percentile is their arithmetic mean. Each dataset has three quartiles, which are the percentiles that divide the dataset into four parts:

- The first quartile is the sample 25th percentile. It divides roughly 25% of the smallest items from the rest of the dataset.
- The second quartile is the sample 50th percentile or the median. Approximately 25% of the items lie between the first and second quartiles and another 25% between the second and third quartiles.
- The third quartile is the sample 75th percentile. It divides roughly 25% of the largest items from the rest of the dataset.

Ranges: The range of data is the difference between the maximum and minimum element in the dataset. You can get it with the function.

1.2.1 Measures of Correlation Between Pairs of Data

You'll often need to examine the relationship between the corresponding elements of two variables in a dataset. Say there are two variables, x and y, with an equal number of elements, n. Let x_1 from x correspond to y_1 from y, x_2 from x to y_2 from y, and so on. You can then say that there are n pairs of corresponding elements: (x_1, y_1), (x_2, y_2), and so on.

You'll see the following **measures of correlation** between pairs of data: (*3*)

- **Positive correlation** exists when larger values of x correspond to larger values of y and vice versa.
- **Negative correlation** exists when larger values of x correspond to smaller values of y and vice versa.
- **Weak or no correlation exists** if there is no such apparent relationship.

The following figure shows examples of negative, weak, and positive correlation:

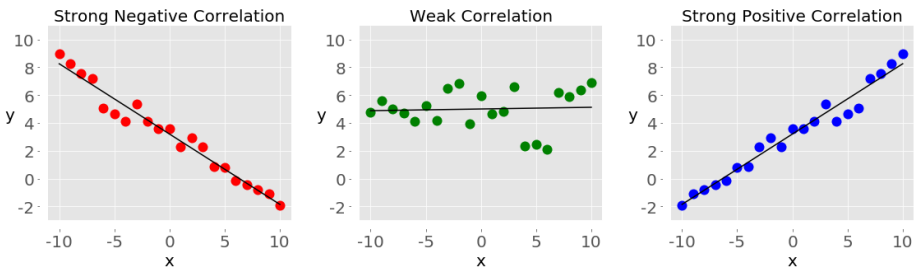


Figure 4 Measures of Correlation

1.2.2 Covariance

The **sample covariance** is a measure that quantifies the strength and direction of a relationship between a pair of variables:

- **If the correlation is positive,** then the covariance is positive, as well. A stronger relationship corresponds to a higher value of the covariance. It shows strong relation with global active power and global intensity as shown in the figure.
- **If the correlation is negative,** then the covariance is negative, as well. A stronger relationship corresponds to a lower (or higher absolute) value of the covariance.
- **If the correlation is weak,** then the covariance is close to zero.

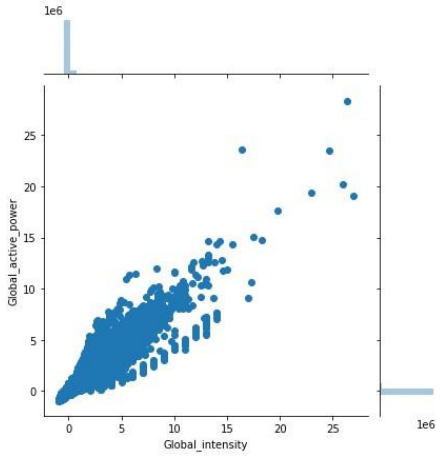


Figure 5 Covariance between Global active power and Global intensity

1.2.3 Correlation Coefficient

The **correlation coefficient**, or **Pearson product-moment correlation coefficient**, is denoted by the symbol r. The coefficient is another measure of the correlation between data. You can think of it as a standardized covariance. Here are some important facts about it: (*3*)

- **The value $r > 0$** indicates positive correlation.
- **The value $r < 0$** indicates negative correlation.
- **The value $r = 1$** is the maximum possible value of r. It corresponds to a perfect positive linear relationship between variables.
- **The value $r = -1$** is the minimum possible value of r. It corresponds to a perfect negative linear relationship between variables.
- **The value $r \approx 0$**, or when r is around zero, means that the correlation between variables is weak.

1.2.4 Visualizing Data

In addition to calculating the numerical quantities like mean, median, or variance, you can use visual methods to present, describe, and summarize data. In this section, you'll learn how to present your data visually using the following graphs:

- Box plots
- Histograms
- Pie charts
- Bar charts
- X-Y plots
- Heatmaps

matplotlib.pyplot is a very convenient and widely-used library, though it's not the only Python library available for this purpose. By using all these function, we can visualize the data in explanatory analysis.

1.2.5 Descriptive Statistics for time series data

The three major components of time series data are:

Trend (T) is a long-term increase or decrease in the data which can assume a great variety of patterns (e.g., linear, exponential, damped, and polynomial).

Real time series with an increasing trend can be found in phenomena related to the demographic development, gradual change of consumption habits, and demand for technologies in the social sectors.

The decreasing trend, in turn, can be found in series concerning the mortality rates, epidemics, and unemployment.

Seasonality (S) is the occurrence of cyclic patterns of variation that repeat, at relatively constant time intervals, along with the trend component. Examples of seasonal patterns are the increase in sales of air conditioners in summer and warm clothing in winter.

Residue (R) is the short-term fluctuations that are neither systematic nor predictable. In the real world, unforeseen events cause such instabilities, such as natural disasters, terrorist attacks, and strikes. (*4*)

1.3 Exploratory Analysis

This section contains some exploratory analysis of the data set used in this project and highlights the important characteristics found. Data exploration consists of a preliminary examination of the data, which is important for understanding the type of information that has been collected and what they mean. In combination with the information acquired during the definition problem, this categorization will determine which method of data analysis will be most suitable for arriving at a model definition. (*5*)

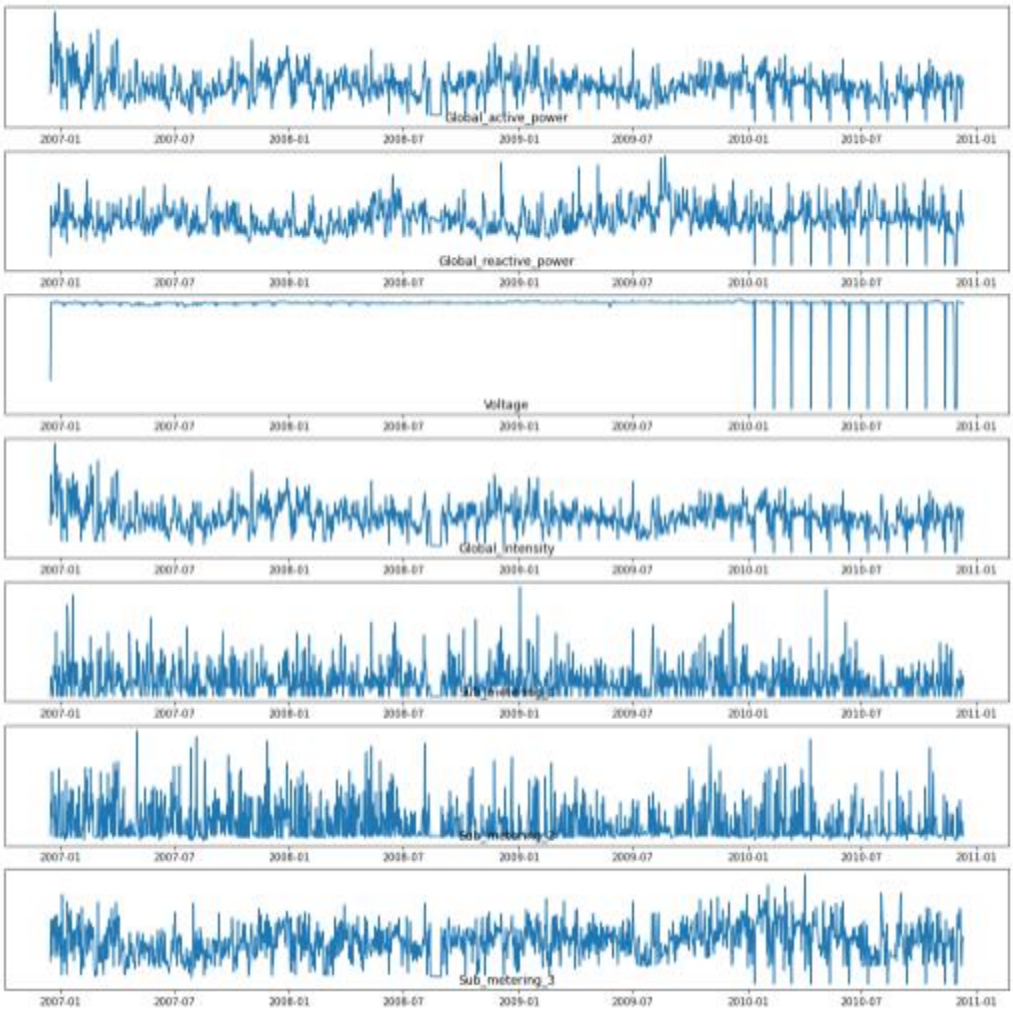


Figure 6 Visualization for all the attributes for 4 years

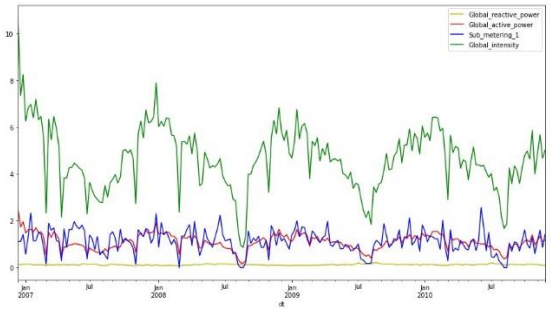


Figure 7 Visualization for a week resampled for all the attributes for 4 years

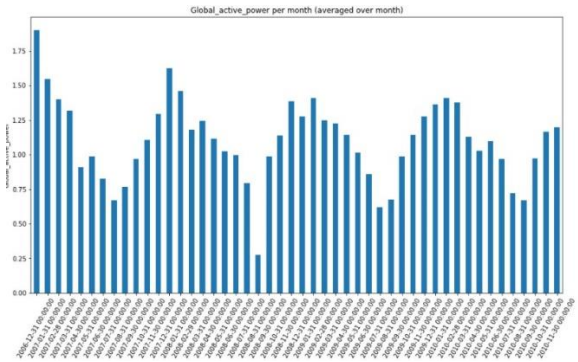


Figure 8 Visualization for Global active power resampled over month for 4 years

The visualization is made on the following criteria, Global active power is considered as the feature Because it is showing characteristics of the stationary data and not showing the trend, seasonality, residue, and the other attributes were also part of it. The 6 visualizations made is shown below in the table. (*6*)

1	Average power consumption for each day in 4 years
2	Average power consumption for each month
3	Average power consumption for each day in a month
4	Average power consumption for each day in a week
5	Average power consumption for each hour in a day
6	Average power consumption for each month in 4 years

1.3.1.1 *Average power consumption for each day in 4 years*

The average power consumption for each day in 4 years is visualized shown below

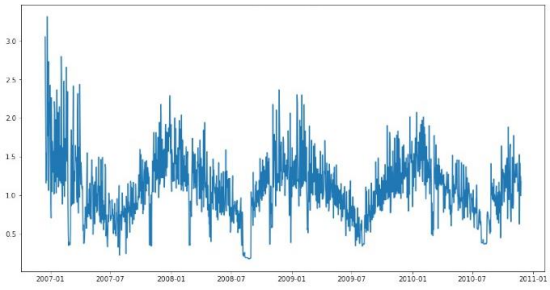


Figure 9 Average power consumption for each day in 4 years

The above graph shows the average power consumption for each day in for the duration of 4 years. The horizontal axis depicts the time period of 4 years starting from 01/2007 to 12/2010 through which the energy consumed was observed, and the vertical axis depicts the number of kilowatt of energy consumed over the time. From the line graph, we can tell that there is a demand for power can be observed in the month of January and in the month of December. The minimum power consumed in the household is in the month of August of every year and there is rise in the power consumption start from September till the end of January. Then there is fall in power consumption starting from January until end of the September. Whereas with the maximum power consumed

there are variations observed and the highest power consumed is shown in the year 2010 with approximately 3k Wh and the lowest power consumed is shown in the year 2008 with approximately 0.5 k Wh shown in the figure.

1.3.1.2 *Average power consumption for each month*

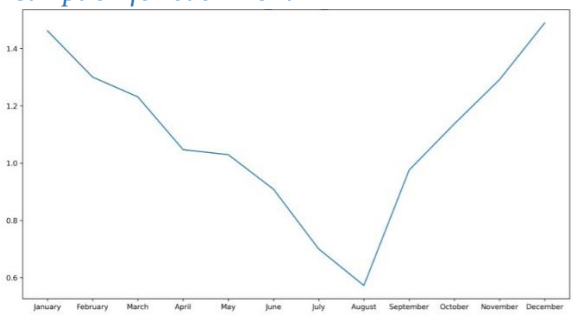


Figure 10 Average power consumption for each month

The above graph shows the average power consumption for each month in for the duration of 4 years. The horizontal axis depicts the time period of months starting from January until the December and the vertical axis depicts the number of kilowatt of energy consumed over the time. From the line graph, we can tell that there is a demand for power can be observed in the month of January and in the month of December. The minimum power consumed in the household is in the month of August year and there is rise in the power consumption start from September till the end of January. Then there is fall in power consumption starting from January until end of the September. Whereas with the maximum power consumed there are variations observed and the highest power consumed is shown in the month and January and December which is average of 1.5 k Wh of Power.

1.3.1.3 *Average power consumption for each day in a month*

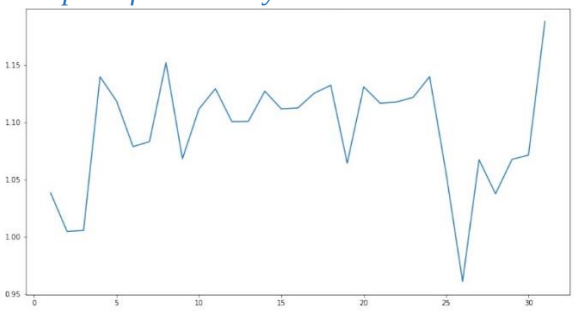


Figure 11 Average power consumption for each day in a month

The above graph shows the average consumption for each day in a month. The horizontal axis depicts the days in a month through which the power consumption was observed, and the vertical axis depicts the number of k Wh of Power consumed over the month.

1.3.1.4 *Average power consumption for each day in a week*

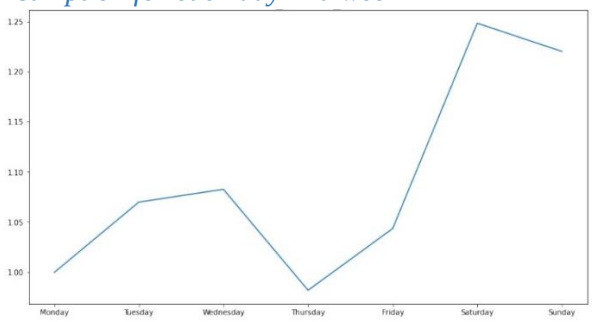


Figure 12 Average power consumption for each day in a week

The above graph shows the weekly profile of power consumed. The horizontal axis depicts the days in a week through which the power consumed was observed, and the vertical axis depicts the number of k Wh of power consumed over the week. From the line graph, we can tell that there is a power consumed can be observed in more in weekends than weekdays. The average minimum power consumed in the household was on Mondays and Thursdays. Then there is rise in power consumption starting from Friday until end of the Saturday. Whereas with the maximum power consumed there are variations observed and the highest power consumed is shown in the Saturday which is average of 1.25 k Wh of Power.

1.3.1.5 *Average power consumption for each hour in a day*

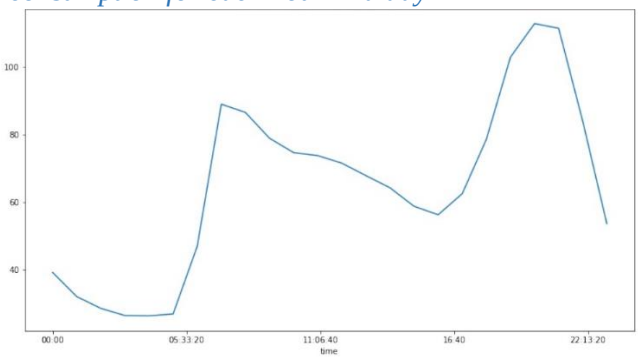


Figure 13 Average power consumption for each hour in a day

The above graph shows the average daily profile of power consumed. The horizontal axis depicts the hours in a day through which the power consumed was observed, and the vertical axis depicts the number of k Wh of power consumed over the day. From the line graph, we can tell that there is a power consumed can be observed in more in evening time and night time starting from 16:00 hour till 22:00 hour. The average minimum power consumed in the household was during the night time. Then there is rise in power consumption starting from 5:00 hour until the 10:00 hour end because of the usage of kitchen and shower. Whereas with the maximum power consumed there are variations observed and the highest power consumed is shown in the 21:00 hour 100 k Wh.

1.3.1.6 *Average power consumption for each month in 4 years*

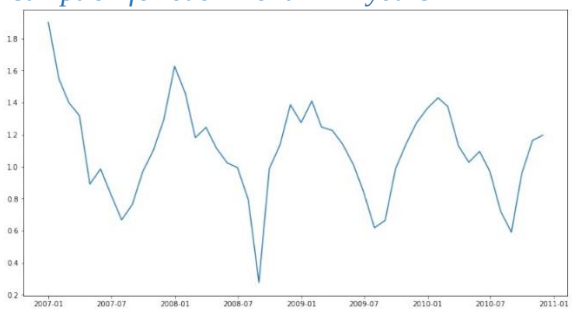


Figure 14 Average power consumption for each month in 4 years

The above graph shows the average power consumption for month day in for the duration of 4 years. The horizontal axis depicts the time period of 4 years starting from 01/2007 to 12/2010 through which the energy consumed was observed, and the vertical axis depicts the number of kilowatt of energy consumed over the time. From the line graph, we can tell that there is a demand for power can be observed in the month of January and in the month of December. The minimum power consumed in the household is in the month of August of every year and there is rise in the power consumption start from September till the end of January. Then there is fall in power consumption starting from January until end of the September. Whereas with the maximum power consumed there are variations observed and the highest power consumed is shown in the year 2010 with approximately 3k Wh and the lowest power consumed is shown in the year 2008 with approximately 0.5 k Wh.

1.3.1.7 *Power consumptions in seasons:*

Power consumption in seasons, here we are considering sub meter 3 as a feature to visualize usage of the heaters during seasons. The average consumption for seasons is explained below:

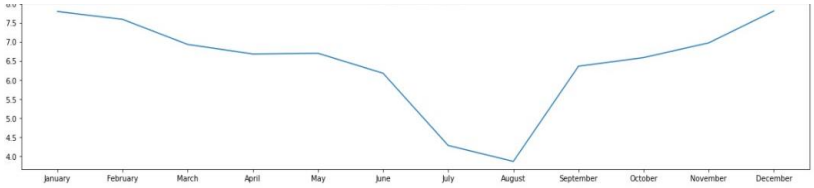


Figure 15 Power consumptions over seasons

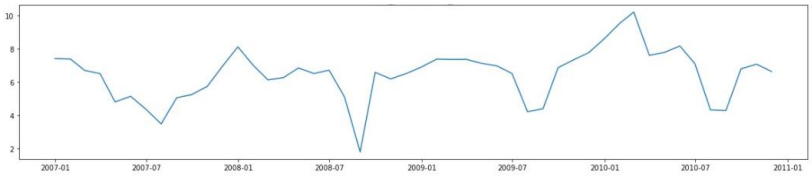


Figure 16 Power consumptions in seasons for 4 years

A brief dip in demand in the tertiary sector and industry late in December, when economic activity slows due to the year-end holidays. Decreases are also seen in both sectors during school holidays (in August, for example). The above line graph depicts the average weekly demand at reference temperatures for a typical July to June period for the residential sector. The horizontal axis shows the months starting from July to June in a year, and the vertical axis shows the Megawatt of energy

in demand for the residential sector. From the graph, we can say that there was a continuous increase in the demand for the energy from September till January but after January there is a continuous decrease from April till June. The highest demand is approximately 28000 Megawatt was recorded in January and the lowest demand of energy temperature was recorded and shared between June to July with approximately 10500 Megawatt of energy (11)

1.4 Algorithm

The step by step approach for building LSTM model for prediction is explained in the below section. (*7*)

1.4.1 LSTM Algorithm

There are several steps involved:

Step 1 : Importing the libraries required to perform LSTM model

```
## These are the liberaries used to run LSTM Model.
## Keras lib provides accruate results in Machine learning. LSTM is the predictive model
## MinMaxScaler is for transform the attribute values in between 0 and 1
## RMSE is used for evaluate the error

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from keras.layers import Dense
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers import Dropout
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
```

Step 2: Scaling the train and test dataset using MinMaxScaler

```
## Normalize features
scaler = MinMaxScaler(feature_range=(0, 1))
```

Step 3: Creating LSTM model for training

```
## design network
model = Sequential()
model.add(LSTM(100, input_shape=(train_x.shape[1], train_x.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
```

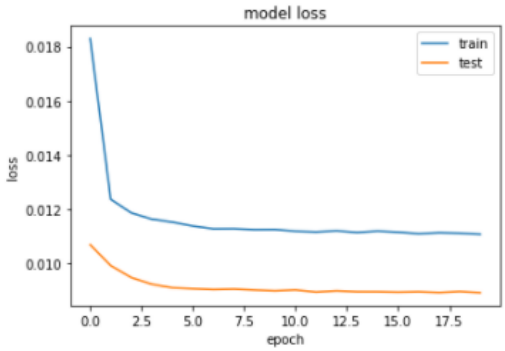
Step 4: Model fit function

```
# fit network
history = model.fit(train_x, train_y, epochs=20, batch_size=70, validation_data=(test_x, test_y), verbose=2, shuffle=False)
```

```
Epoch 1/20
126/126 - 2s - loss: 0.0183 - val_loss: 0.0107
Epoch 2/20
126/126 - 1s - loss: 0.0124 - val_loss: 0.0099
Epoch 3/20
126/126 - 1s - loss: 0.0119 - val_loss: 0.0095
Epoch 4/20
126/126 - 1s - loss: 0.0116 - val_loss: 0.0092
Epoch 5/20
126/126 - 1s - loss: 0.0115 - val_loss: 0.0091
Epoch 6/20
126/126 - 1s - loss: 0.0114 - val_loss: 0.0091
Epoch 7/20
126/126 - 1s - loss: 0.0113 - val_loss: 0.0090
Epoch 8/20
126/126 - 1s - loss: 0.0113 - val_loss: 0.0090
Epoch 9/20
126/126 - 1s - loss: 0.0112 - val_loss: 0.0090
Epoch 10/20
126/126 - 1s - loss: 0.0112 - val_loss: 0.0090
Epoch 11/20
126/126 - 1s - loss: 0.0112 - val_loss: 0.0090
Epoch 12/20
126/126 - 1s - loss: 0.0112 - val_loss: 0.0089
Epoch 13/20
126/126 - 1s - loss: 0.0112 - val_loss: 0.0090
Epoch 14/20
126/126 - 1s - loss: 0.0111 - val_loss: 0.0089
Epoch 15/20
126/126 - 1s - loss: 0.0112 - val_loss: 0.0089
Epoch 16/20
126/126 - 1s - loss: 0.0111 - val_loss: 0.0089
Epoch 17/20
126/126 - 1s - loss: 0.0111 - val_loss: 0.0089
Epoch 18/20
126/126 - 1s - loss: 0.0111 - val_loss: 0.0089
Epoch 19/20
126/126 - 1s - loss: 0.0111 - val_loss: 0.0090
Epoch 20/20
126/126 - 1s - loss: 0.0111 - val_loss: 0.0089
```

Step 4: Plotting the losses

```
# plot history
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper right')
plt.show()
```



Step 5: Scaling the predictions

```
# make a prediction
yhat = model.predict(test_x)
test_x = test_x.reshape((test_x.shape[0], 7))
```

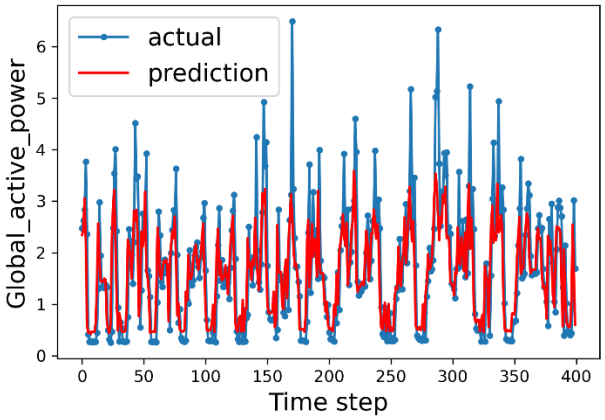
Step 6: Inverse transform of scaled data

```
# invert scaling for forecast
inv_yhat = np.concatenate((yhat, test_x[:, -6:]), axis=1)
inv_yhat = scaler.inverse_transform(inv_yhat)
inv_yhat = inv_yhat[:,0]

# invert scaling for actual
test_y = test_y.reshape((len(test_y), 1))
inv_y = np.concatenate((test_y, test_x[:, -6:]), axis=1)
inv_y = scaler.inverse_transform(inv_y)
inv_y = inv_y[:,0]
```

Step 7: Plotting the test data against the actual

```
px=[x for x in range(400)]
plt.plot(px, inv_y[:400], marker='.', label="actual")
plt.plot(px, inv_yhat[:400], 'r', label="prediction")
plt.ylabel('Global_active_power', size=15)
plt.xlabel('Time step', size=15)
plt.legend(fontsize=15)
plt.show()
```



Step 8: Calculating the losses

```
# calculate RMSE
rmse = np.sqrt(mean_squared_error(inv_y, inv_yhat))
print('Test RMSE: %.3f' % rmse)
```

The Thesis experimental setup has explained in next chapter.

2 References

1. *UCI Machine Learning Repository: Individual household electric power consumption Data Set*. https://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption (accessed 2020-11-26).
2. Wikipedia. *AC power*. https://en.wikipedia.org/w/index.php?title=AC_power&oldid=984930043 (accessed 2020-11-26).
3. Python, R. Python Statistics Fundamentals: How to Describe Your Data. *Real Python*, Dec 16, 2019. https://realpython.com/python-statistics/ (accessed 2020-11-26).
4. Parmezan, A. R. S.; Souza, V. M.; Batista, G. E. Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. *Information Sciences* **2019,** *484*, 302–337. DOI: 10.1016/j.ins.2019.01.076.
5. GitHub. *Pushpak2227/Master-Thesis*. https://github.com/Pushpak2227/Master-Thesis/blob/master/EDA%20with%20data%20cleaning%20and%20visualizations.ipynb (accessed 2020-11-26).
6. GitHub. *Pushpak2227/Master-Thesis*. https://github.com/Pushpak2227/Master-Thesis/blob/master/Visualizations%20for%20a%20day%2C%20week%2C%20month.ipynb (accessed 2020-11-26).
7. GitHub. *Pushpak2227/Master-Thesis*. https://github.com/Pushpak2227/Master-Thesis/blob/master/predicting%20the%20power%20consumption%20for%20the%20next%20hour%20based%20on%20the%20power%20consumption%20over%20last%2024%20hrs.%20.ipynb (accessed 2020-11-26).