Technical University of Berlin

Data analysis for predicting the energy consumption in an household for middle European climatic conditions

Master Thesis

Submitted in partial fulfilment of the requirements for the academic degree of

Master of Science

IT for Energy

Submitted by: Pushpak Hanabe Prakashkumar

Mat No: ITE- 404397

First Supervisor: Prof. Dr Tetyana Morozyuk

Second Supervisor: Prof. Alberto Diaz-Durana

Berlin, December 2020

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Ausführungen, die anderen veröffentlichten oder nicht veröffentlichten Schriften wörtlich oder sinngemäß entnommen wurden, habe ich kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Fassung noch keiner anderen Prüfungsbehörde vorgelegen.

Berlin, den ______________________________

(Pushpak Hanabe Prakashkumar)

ACKNOWLEDGEMENTS

Firstly, I would like to thank you, Prof. Dr Tetyana Morozyuk, for the opportunity to work on such an exciting topic and for several helpful discussions in the thesis. I came to you when I didn't find the right supervisor to support the topic. You provided confidence to go with the topic which I wanted to do itself, from choosing the thesis title and providing me with the right second supervisor for the topic, and thank you so much for everything.

Secondly, I would like to thank Prof. Alberto Díaz-Durana for being the second supervisor, for supporting me with everything I've done so far. From the beginning with the thesis structure and until the results are achieved. You taught me what to write in the dissertation, and you wanted me to learn more for my goodness. You have shown the best way to finding solutions to the problems that so I am grateful to you, from the bottom of my heart.

I also would like to thank Prof. Francesca Bugiotti and Prof Rakan Razouk for teaching the Data analysis to implement in the energy sector.

Lastly, I want to say thank to friends and family for being loyal and always supporting.

Berlin, December 2020

Pushpak Hanabe Prakashkumar

ABSTRACT

The aim and objective of this thesis are to build an LSTM model to forecast the electric power consumption in a household and to find the most suitable forecasting period for days in a week and to analyze the power consumptions during several periods and seasons. The time-series data in our thesis was on a household. The dataset has electricity consumption from December 2006 to November 2010. The data analysis has been performed with the deep neural networks, LSTM (Long short term memory) model and to explore and understand the dataset; In the thesis, line and histograms plots are used for series data for the data distribution. The suitable forecasting method and the most suitable forecasting period were chosen by considering the smallest value of RMSE (Root Mean Square Error). The concluded result of the thesis showed that the LSTM model was the accurate model for finding the forecasting for an hour. On the other hand, LSTM model showed that it is not easy to forecast on the weekends like Saturday compared to weekdays like Tuesdays and Fridays.

Keywords: Power consumption forecast, LSTM model, Deep learning with Keras.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Nomenclature

Abbreviations

LSTM	Long Short-Term Memory
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
KNN	K-Nearest Neighbors
BPNN	Back-Propagation Neural Network
KRR	Kernel Ridge Regression
GPR	Gaussian Process Regression
SVR	Support Vector Regression
SVM	Support Vector Machine
RNN	Recurrent Neural Network
ARIMA	Autoregressive Integrated Moving Average
DNN	Deep Neural Network
AI	Artificial Intelligence
DL	Deep Learning
ML	Machine Learning
RELU	Rectified Linear Unit
MSE	Mean Squared Error
NLP	Natural Language Processing
CNN	Convolutional Neural Network
ADAM	Adaptive Moment Estimation
ARMA	Autoregressive Moving Average
EE LSTM	Evolutionary Ensemble Long Short-Term Memory

Symbols

$\mathbb{U}$	LSTM Unit
h	Hidden state
c	Cell state
b	Bias
w	Weight vector
f_g	Forget gate
i_g	Input gate
o_g	Output gate
$\mathbb{R}$	Recurrent network
$\mathcal{L}$	layer
$\mathcal{S}$	Sequence
$\mathcal{F}$	Function
t	Timestamps
σ	Sigma function
p	Percentile
r	Correlation Coeffient
KW	Kilowatt

1 INTRODUCTION

Humans use energy in several different forms in order to run machinery and accomplish different tasks [1, p. 7]. The demand for power energy is affected by consumers behaviour and their use of electrical appliances. Therefore, the power grid authorities are considering the need to develop and invent new ways of efficiently handling the consumption of power in industrial and residential buildings that will regulate energy demand[2, p. 123369].

Electricity load forecasting has now gained significant significance in modern electrical power management systems with elements of smart greed technology. A reliable forecast of the consumption of electricity is a starting point for policymaking and progress in the production and distribution of energy. At the level of individual households, the capability to effectively forecast electricity usage dramatically reduces prices using suitable energy storage systems. Consequently, future energy-efficient power grids would need entirely new ways of predicting demand on the scale of individual households. [3].

1.1 Background

The average French household consumes about 4760 kWh of electricity per year [4]. However, specific energy consumption depends on energy usage and accommodation. Household consumption has increased over the previous decades with the rise in electricity-hungry consumer appliances. Electricity is what most individuals in industrialised regions of the world use every day [1, p. 7]. Nevertheless, when outages happen, business owners and consumers alike deal with inconveniences ranging from spoiled food to productivity losses. People see their expected energy usage for the week or month, how small changes might allow them to save resources[5], Hsiao et al. proposed a model on household electricity demand forecast on context information and user daily schedule analysis from meter data [6, p. 33]. To make predictions from the smart meter data, we have issues regarding forecasting as short-term, medium-term and long-term. Thesis studies about Short-term forecasting issues include forecasting behaviour for only a few time frames (days, weeks or months) in the future[7]. Short-and medium-term forecasting is usually focused on the detection, modelling and extrapolation of trends found in historical data. Since these historical data typically show inertia and do not alter significantly high-speed, Machine learning techniques and Deep learning are useful for short-and medium-term forecasting [8].

1.1.1 Thesis context

The thesis will predict the electricity of a household from the previously recorded values and evaluate the prediction model with the deep neural network, LSTM. Thesis benefits students to learn the deep learning methods for predicting effectively and improve their prediction. Additionally, the thesis gives a brief overview of time series prediction using deep neural network, LSTM model. The thesis explains exploratory data analysis and consumptions in several periods of the year and seasons on the dataset to visualise the power consumption data with various patterns.

1.2 Problem

The problem considered by this thesis is predicting the average electricity consumption for a household in France. Studies have found that as the forecasting time increases, the data become more structured and challenging to solve with simple machine learning algorithms. Accuracy will be less so neural networks are used because it is shown in previous results. This thesis is an inquiry into how a selection is made on Deep learning algorithms work on time series data set. Mainly, the thesis deals with the following questions:

- How to analyse and visualise the power consumption data in different period?
- How to build forecast models for small time steps for an hour and a week using LSTM?
- How accurate the LSTM model for the forecasting?

Additionally, this thesis intends to provide a visualisation of power consumption in different period.

1.3 Purpose

The purpose of the thesis is to present a theoretical and practical investigation into the use of LSTM Deeping learning method for forecasting the average electricity consumption for a household. The purpose is to determine how effective is the prediction and to evaluate how beneficial it is for the electricity forecasting for an hour and a week.

1.4 Motivation

The motivation of the thesis will help the electricity producers in peak demand to ensure that they have enough electricity for a household in its peak demand (4). It also benefits the researchers to develop different methods for prediction for time series data using Artificial neural networks. It also describes the overview of Deep neural networks learning model and exploring dataset power consumption dataset and to see their features and patterns of consumptions during the period, and prediction through LSTM model.

1.4.1 Benefits

This thesis offers a theoretical strategy for implementing the knowledge of deep learning in the electricity sector to have sufficient electricity for the benefit of their electric power forecasting. Secondarily, the method will enable ideas to other researchers working in the electricity sector for the forecasting. This research is carried out using a fixed real data set from a single point in time, but for thesis purposes. This concept will stay relevant and applicable for a prolonged period. The characteristics of electricity use and trends of household consumption are unlikely to improve in the coming decades, except for a growing trend that shifts too slowly to alter the results of this research.

1.5 Research Methodology

The prediction of electricity consumption includes the analysis and use of quantitative data set, and evaluation of the precision of the prediction are based on quantitative error measures(RMSE), and so on the methodologies used in this work are primarily quantitative. The household power consumption data is collected from open source. The thesis approach would be experimental since it focuses on prediction and causality between dependent and independent variables in the dataset. The thesis time horizon will be cross-sectional, meaning that it is the study of electricity consumption at a particular point in time. Finally, the thesis techniques and procedures will come from the field of Deep learning [9, p. 23]. The work presented by this thesis is an evaluation of the effectiveness of a selection of supervised linear regression method at predicting the electricity consumption of a household. The work of this thesis demonstrates that Recurrent Neural network (LSTM) [10, p. 1521]is the most accurate method within the constraints of the problem considered. In addition to accuracy, the prediction for a week based on several past days values considered.

1.6 Outline

Chapter 2 will provide the Literature Review Chapter 3 will provide the theoretical context of deep learning, and the methods used some theoretical considerations related to time series data. The LSTM model for time series forecasting Chapter 4 methods will include a summary and descriptive overview of the data set and visualisations used and a description of the algorithms used for the experimental section. Chapter 5 contains the LSTM forecasting model, provided with results and discussions of the experiments, along with a summary of the findings. Chapter 6 will provide Conclusions and future work.

2 LITERATURE REVIEW

Previous researchers have aimed to forecast power consumption by analysing the statistical and machine learning models. [11] , [12]. In specific, neural network-based predictions can be effectively used to learn patterns of electrical consumption using time series data.

Nogales et al. proposed an LSTM-based model to learn trends of electricity use for individual households as well as for various households[13], They have three approaches (naive, auto arima, LSTM model). They considered MAE to calculate the Error. The results shown that the LSTM approach is better than all, shows 19% better than auto arima.

Kong et al. proposed Short term load forecasting based LSTM Recurrent network [14] made predictions under multiple scenarios with different time horizons using the LSTM model and compared it with other models including BPNN, k-nearest neighbors algorithm (KNN), and ELM. Kong et al. addressed this problem at the substation and household level. They compared the prediction performance of two cases: aggregating the individual load prediction and predicting the aggregate load at the substation level. The result shows that, by aggregating the individual load prediction data, the performance improved by 0.49% and 1.08% for LSTM and BPNN-T models, respectively.

Linton proposed a model to predict hourly electricity for households using kernel machine learning models [12]. They have used several kernel models (KNN, KRR, GPR, SVR) for several set of households. They found that KNN gives promising results for small size dataset. They found that SVM that GPR model performed well for a large set of 64 households. They have also recommended making use of Neural network model to see good results.

Chakravorty et al. proposed Evolutionary Ensemble LSTM based Household Peak Demand Prediction [15]. They predicted for next minute peak demand. They have compared using two LSTM models, such as LSTM and EE LSTM. They found the results that EE LSTM performs well both for cluster network or cluster network.

Marvuglia et al. proposed forecasting electricity of a household using RNN [16]. They have used the RNN model for predicting one hour, and They achieved error value of 1.5% for test week and 4.5% of Error for the mean.

Parade proposed Individual a household prediction using the ARIMA model [17]. This study used ARIMA model for prediction for a household for a week and error value as RMSE and got overall RMSE around 465 Kilowatts.

The result shows the promising advantages of LSTM approach in the challenging short-term electric load forecasting field. However, most of the prediction approaches conducted at households [12, 13, 15, 17]. Multiple writers have presented several approaches based on different criteria, which are of interest to research on the LSTM model. The research has shown that LSTM offers a promising forecast of power demand. To the authors best knowledge, They have some drawback, and they have not presented an Exploratory study of data over time. However, This study presents a multi-period data analysis, statistics and an hour and a weekly forecast of LSTM model. In this thesis, Root mean squared error (RMSE) will be used as evaluation metrics to compare the observed values and the predicted one. It works better to analyse the influence of LSTM models rather than introducing a new predictive model to improve higher predictive accuracy than other models; this paper aims to determine the impact of the LSTM model on prediction performance. The experimental setup in this thesis is conducted using Vanilla LSTM models without fine-tuning but maintaining a specific neural network specification.

3 THEORETICAL BACKGROUND

This chapter provides the theory behind the household power consumption, Deep learning method methods and the components of the DNN and Time series data; It provides a complete theoretical explanation on the LSTM model.

3.1 Household power consumption

The household energy consumed data is collected from an open-source. It is explored in Jupyter notebook(Open source platform to view the recordings listed in the dataset), which are the global active power, global reactive power, voltage, intensity, submetering 1, submetering 2 and submetering 3. Then it is explored to observe the timestamps of power consumed in periods.

The global active power is averaged over a full period of the AC waveform, lands in a net energy transfer in one direction are known to be active power in kilowatt. The global reactive power is the power retained, which returns to the source in each phase of the household, which is measured in kilowatt. Voltage is that the tension from the power source of the electrical circuit, which moves the charged electrons (current) into the conductive loop, allows them to undertake to work out how to illuminate a lightweight circuit. Intensity is that the speed, per unit time, at which a circuit transfers current measured in kilowatt. The final categories are the power consumption in submetering 1, which average power consumption is consisting mainly of a dishwasher, an oven and a microwave. The submetering 2 shows the readings from the laundry room, which includes a washing machine, a tumble dryer, a refrigerator and a light. The submetering 3 consists of the average power consumption of an electric water heater and an air-conditioner.

For a household, electricity consumption is highly stochastic due to the active load. The below figure 3-1 shows the visualization of several attributes.

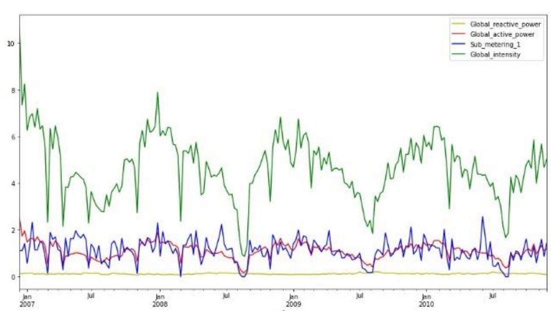


Figure 3-1 Pattern visualisation for several attributes in the dataset

The thesis is more obsessed with the prediction; the knowledge of machine learning can provide the science for making it possible. This thesis is explicitly chosen deep learning for the prediction of power consumption. The following sections explain Deep learning, Deep neural network frameworks, and the components of DNNs and Time-series data. At last, it provides a theoretical framework for the LSTM model

3.2 Deep learning

DL is a sub-field of ML in artificial intelligence (AI) that deals with algorithms influenced by the biological structure and functioning of the brain to help machines [18]. The deep learning can be supervised, semi-supervised or unsupervised. The implementation of deep learning and regression is discussed in the thesis. [19]

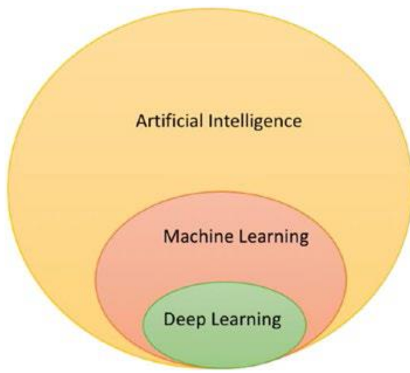


Figure 3-2 Branches of Artificial Intelligence [20]

Bringing all together, we can say that AI is the field of inducing intelligence to an artificial computer or device, with or without explicit programming. ML is a subfield in AI where intelligence is induced without explicit programming. Lastly, DL is a field within ML where intelligence is induced into systems without explicit programming using algorithms that have been inspired by the biological functioning of the human brain.

The increased number of researchers and practitioners in the AI field and have created a mature and benevolent community. Today, it is reasonably easy to access tools, research papers, datasets, and in fact, even infrastructure to practice DL as a field. For our first use case, we would need a dataset and a problem to get started. The open sources to download the dataset are provided below.

• Kaggle: www.kaggle.com

• US Government Open Data: www.data.gov

• Indian Government Open Data: https://data.gov.in

• Amazon Web Service Datasets: https://registry.opendata.aws

• Google Dataset Search: https://toolbox.google.com/datasetsearch

• UCI ML Repository: https://archive.ics.uci.edu/ml

Deep learning frameworks

Deep learning frameworks provide building blocks for the architecture, training and validation of deep neural networks through a high-level programming interface[21]. It is categorised into two types: they are Low-Level DL Frameworks (Theano, PyTorch, MxNET, TensorFlow) and High-Level DL Frameworks (Keras). This Keras framework is part of our thesis because Keras is a popular and easy-to-use open-source Python library for designing and testing deep learning models. It wraps up the powerful numerical computing libraries Theano and TensorFlow and allows to define and train neural network models for just a few lines of code. [22]

Deep Neural Networks

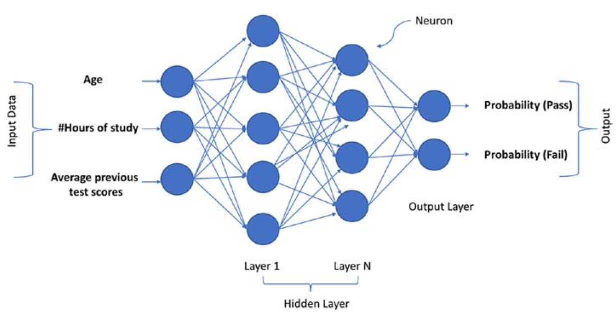


Figure 3-3 A Multiple layer Neural network[20]

DNN has two hidden layers with five and four neurons, respectively. The first hidden layer accepts input data that has three dimensions and gives an output in the output layer with two neurons. To have this make intuitive sense, we can assume that this is a simple DNN for a problem like predicting whether a student will pass or fail based on some input data. Say we have the age, the number of hours studied, and the average score out of 100 for all the previous tests for which he appeared as the input data point. Building a neural network in Keras is as simple as the following script. Just for the above problem, we are considering three steps to achieve the goal.

- Getting the Data Ready
- Defining the Model Structure
- Training the Model and Making Predictions

Those steps are possible only when the science behind the DNN understood first. The implementation of DNNs is crucial to a simple problem above or thesis problem, So the below section explained briefly about the components of the DNNs.

3.2.1 Components of DNN

Neuron: At the core of the DNN, we have neurons where computation for output is executed. A neuron receives one or more inputs from the neurons in the previous layer. If the neurons are in the first hidden layer, they will receive the data from the input data stream

It is necessary to have a function that operates on the sum of the input multiplied by the corresponding weights and reacts with an appropriate value based on the input If a higher impact input is provided, the output should be higher, and vice versa. It is parallel to the activation signal (i.e. more substantial influence-> then trigger, otherwise deactivate) [20]

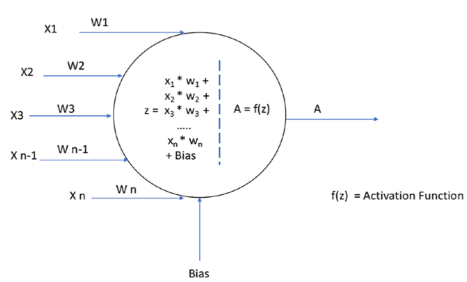


Figure 3-4 A single Neuron[20]

Activation Function: It is the function that takes the combined input z as shown in the preceding illustration, applies the function to it and transfers the output value, attempting to imitate the Activate/Deactivate function. As a consequence, the activation function decides the state of the neuron by measuring the activation function on the combined input. [20] The most common one is the ReLU (rectified linear unit), ReLU activation function is considered for the thesis.

ReLU Activation Function

The ReLU uses the function $f(z) = \max(0, z)$, which means that if the output is positive it would output the same value, otherwise it would output 0. The function's output range is shown in the following visual.

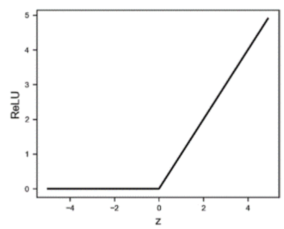


Figure 3-5 ReLU activation function[20]

The function may look linear, but it is not. ReLU is a proper nonlinear function and in fact, works well as an activation function. It not only improves the performance but significantly helps the number of computations to be reduced during the training phase[20].

Model: The overall configuration of the DNN is based on the model object in Keras, which offers a simple way to build a layer stack by inserting new layers one after the other. The simplest way to describe a model is to use a sequential model to construct a linear stack of layers easily.

Layers: A layer in the DNN is defined as a group of neurons or a logically separated group within a hierarchical network structure. There are a few significant layers that we can use in most cases.

Dense Layer*:* It is layer is a regular layer of DNN that connects every neuron in the given layer to every neuron in the previous layer. [20]

Dropout Layer: It is layer in DL helps reduce overfitting by introducing regularisation and generalisation capabilities into the model. In the literal sense, the dropout layer drops out a few neurons or sets them to 0 and reduces computation in the training process. The process of arbitrarily dropping neurons works quite well in reducing overfitting.

Loss function: The loss function is the metric that helps a network understand whether it is learning in the right direction. There are some popular loss functions available:

- Mean Squared Error - Average squared difference between the actual and predicted value. The squared difference makes it easy to penalise the model more for a higher difference.

$$MSE = \sum_{n=1}^{k} \frac{(Actual - Predicted)^2}{k}$$

$$RMSE = \sum_{n=1}^{k} \sqrt{\frac{(Actual - Predicted)^2}{k}}$$

- Mean Absolute Error – The average absolute error between actual and predicted.

$$MAE = \sum_{n=1}^{k} Actual - predcited$$

Optimisers: The essential part of the model training is an optimiser. There are many optimisers, but Adam optimiser is the most common and commonly used optimiser in DL.

Adam stands for Adaptive Moment Estimation. In some instances, we will blindly pick Adam optimiser and forget about optimisation alternatives. This optimisation technique computes an adaptive learning rate for each parameter. It defines momentum and variance of the gradient of the loss and leverages a combined effect to update the weight parameters. The momentum and variance together help smooth the learning curve and effectively improve the learning process. [20]

$$\boldsymbol{Weights} = \boldsymbol{Weights} - (\boldsymbol{Momemtum\ and\ variance\ combined})$$

Model Configuration: Once designed the network, Keras provides with an easy one-step model configuration process with the 'compile' command. To compile a model, we need to provide three parameters: an optimisation function, a loss function, and a metric for the model to measure performance on the validation dataset. In this section, we have studied the DNN components and why it is used, Now we look into the Times series components in the next section.

3.3 Time series analysis

The time series analysis is the method to analyse the time-series data concerning to extract the meaning of full statistics and other characteristics of the data. The three significant components to analyse the 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). For instance, 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 a 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. [23, p. 4]

That is all about the components of DNN and time series analysis. We have considered a neural network for the time series prediction because Neural Networks is one of the most used machine learning algorithms at present [24]. Neural networks are increasingly becoming more and more like linear regression for statisticians was for data scientists or machine-learning practitioners[25]. It has proven over time that it performs well with accuracy and speed compared to other algorithms, it has various variants like RNN(Recurrent Neural Networks), CNN(Convoluted Neural Networks), AutoEncoder and many more. This thesis explains about RNNs LSTM algorithm specifically for predictions in the next section, and we have used LSTM particularly for the prediction of the power consumption.

3.4 Recurrent neural networks

RNNs are standard models which have shown great promise for many NLP (Natural language processing) tasks. However, despite their recent popularity, In a recent article, we have only found a small number of resources that clarify thoroughly, how RNNs function and how to incorporate them[26]. The concept behind RNNs is to use sequential data. In a typical neural network, we believe that all inputs (and outputs) are independent of each other; however, for many tasks, that is a terrible idea. To determine the next word in a sentence, we learn all about the words that came before it. RNNs are considered recurrent since they serve the same purpose for each sequence variable, the output being dependent on previous computations. RNNs have a "memory" that collects knowledge as to what is being measured. [27].

In several NLP tasks such as Apple Siri and Google voice search, RNNs have shown significant progress. At this point, the most widely used type of RNNs is LSTMs, which are far better at capturing long-term dependencies than RNNs for vanilla[28]. The LSTM model is explained in the next section.

3.4.1 LSTM Model

The structure of a single LSTM cell

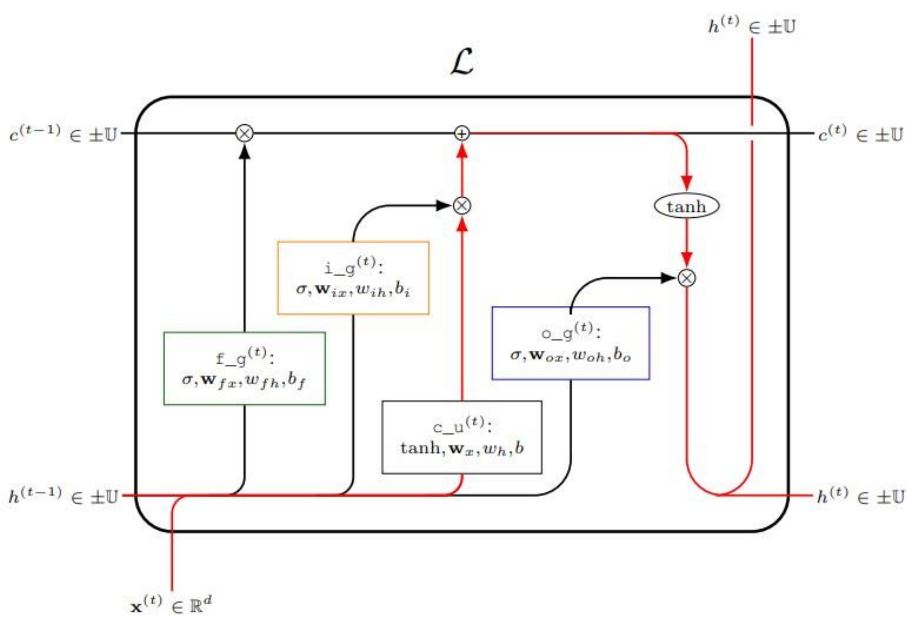


Figure 3-6 Graphic illustrations of a single LSTM cell [29]

The LSTM cell has two tanh activation functions shown in red arrows and has three gates—forget (green), enter (orange) and output (blue)—control the interaction between cell status and hidden state.

Let $U = [0,1]$ represent the unit interval and let $\pm\mathbb{U} = [-1, 1]$. The LSTM cell has two recurrent functions, h and c respectively called the hidden state and the cell state. The cell, denoted $\boldsymbol{\mathcal{L}}$, is a mathematical function that takes three inputs and produces two outputs:

$$\left(\boldsymbol{h}^{(t)}, \boldsymbol{c}^{(t)}\right) = \boldsymbol{\mathcal{L}}\left(\boldsymbol{h}^{(t-1)}, \boldsymbol{c}^{(t-1)}, \boldsymbol{x}^{(t)}\right), \qquad (\mathbf{1})$$

Where $\boldsymbol{h}^{(t)}, \boldsymbol{h}^{(t-1)}, \boldsymbol{c}^{(t)}, \boldsymbol{c}^{(t-1)} \boldsymbol{\mathcal{E}} \pm \mathbb{U}$ and $\boldsymbol{x}^{(t)} \boldsymbol{\mathcal{E}} \mathbb{R}^d$. At some point (t), both outputs leave the cell, and they are being fed back to the same cell at the time $(t+1)$. At any time point t, an element of the input sequence $\boldsymbol{x}^{(t)} \boldsymbol{\mathcal{E}} \mathbb{R}^d$is also fed into the LSTM cell. Inside LSTM the cell, The hidden state and the input vector are fed to three gates (functions), With the support of a sigmoid activation function, each of which produces a scalar value in U:

$$f\_g^{(t)}(x^{(t)}, h^{(t-1)}) = \sigma\left(w_{f,x}^T, x^{(t)} + \mathscr{w}_{f,h}\, h^{(t-1)} + b_f\right) \mathcal{E}\, \mathbb{U},$$

$$i\_g^{(t)}(x^{(t)}, h^{(t-1)}) = \sigma\left(w_{i,x}^T, x^{(t)} + \mathscr{w}_{i,h}\, h^{(t-1)} + b_i\right) \mathcal{E}\, \mathbb{U},$$

$$o\_g^{(t)}(x^{(t)}, h^{(t-1)}) = \sigma\left(w_{o,x}^T, x^{(t)} + \mathscr{w}_{o,h}\, h^{(t-1)} + b_o\right) \mathcal{E}\, \mathbb{U},$$

Where $w_{f,x}, w_{i,x}, w_{o,x}\, \boldsymbol{\mathcal{E}}\, \mathbb{R}^d$ and $\mathscr{w}_{f,h}, \mathscr{w}_{i,h}, \mathscr{w}_{o,h}, b_f, b_i, b_o\, \boldsymbol{\mathcal{E}}\, \mathbb{R}$, weight vectors and biases, respectively. These are the parameters to be studied during cell training. The three gates can be perceived as switches when their output values are near to 1(on) or 0 (off). Another scalar function, the so-called cell update e (c_u), is constructed as a single neuron with a tanh activation function:

$$c_u{}^{(t)}(x^{(t)}, h^{(t-1)}) = \tanh\left(w_x^T\, x^{(t)} + \mathscr{w}_h h^{(t-1)} + b\right) \mathcal{E}\, \pm \mathbb{U}$$

Where $w_x\, \mathcal{E}\, \mathbb{R}^d$ and $w_h, b\, \mathcal{E}\, \mathbb{R}$ are additional weight parameters to be learned. The forget gate (f_g) is the gate which controls the cell gate to how much to forget cell state. The input gate (i_g) controls how much cell update is inserted to the cell state, and the output gate (o_g) controls how much of the modified cell state should leave the cell and become the next hidden state. At time t, The new cell and hidden states are:

$$c^{(t)} = f\_g^{(t)}\, c^{(t-1)} + i\_g^{(t)}\, c^{(t-1)}\, \mathcal{E} \pm \mathbb{U}$$

$$h^{(t)} = o\_g^{(t)} . \tanh\left(c^{(t)}\right) \mathcal{E} \pm \mathbb{U},$$

where the arguments $(x^{(t)}, h^{(t-1)})$ have been ignored for readability. By designing it using hidden states that move through time, RNNs have the potential to take an input sequence of any size and to generate an output sequence of any size as seen in the graphical representation in Figure 3-7. The user will determine at what time points to feed the input sequence and at what time points to extract the outputs. [30, p. 3]

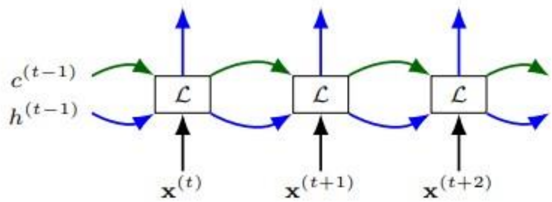


Figure 3-7 The flexibility of an LSTM cell. [29]

A Layer in LSTM cells

A layer of n LSTM cells, which we denote by $\mathcal{L}_n$, corresponds to the concatenation of n cells $\mathcal{L}_1, \mathcal{L}_2, \dots \dots, \mathcal{L}_n$, each one with a different set of internal weights. That is,

$$\left(\boldsymbol{h}_1^{(t)}, \boldsymbol{c}_1^{(t)}\right) = \boldsymbol{\mathcal{L}}\left(\boldsymbol{h}_1^{(t-1)}, \boldsymbol{c}_1^{(t-1)}, \boldsymbol{x}_1^{(t)}\right),$$

$$\left(\boldsymbol{h}_2^{(t)}, \boldsymbol{c}_2^{(t)}\right) = \boldsymbol{\mathcal{L}}\left(\boldsymbol{h}_2^{(t-1)}, \boldsymbol{c}_2^{(t-1)}, \boldsymbol{x}_2^{(t)}\right),$$

$$\vdots$$

$$\vdots$$

$$\left(\boldsymbol{h}_n^{(t)}, \boldsymbol{c}_n^{(t)}\right) = \boldsymbol{\mathcal{L}}\left(\boldsymbol{h}_n^{(t-1)}, \boldsymbol{c}_n^{(t-1)}, \boldsymbol{x}_n^{(t)}\right),$$

which can equivalently be written as

$$\left(\boldsymbol{h}^{(t)}, \boldsymbol{c}^{(t)}\right) = \boldsymbol{\mathcal{L}}\left(\boldsymbol{h}^{(t-1)}, \boldsymbol{c}^{(t-1)}, \boldsymbol{x}^{(t)}\right),$$

Where $\boldsymbol{h}^{(t)}, \boldsymbol{h}^{(t-1)}, \boldsymbol{c}^{(t)}, \boldsymbol{c}^{(t-1)} \boldsymbol{\mathcal{E}} \pm \mathbb{U}$ and $\boldsymbol{x}^{(t)}\ \boldsymbol{\mathcal{E}}\ \mathbb{R}^d$. The biases and individual weight vectors from each of the LSTMs are arranged into matrices. The dot products become matrix-vector products, and scalar multiplications become elementary multiplications. The activation functions are implemented element-wise, enabling the simultaneous evaluation over entire LSTM cells. The cell update function and three gates now have weight matrices $W_{fx}, W_{ix}, W_{ox}, W_x\ \mathcal{E}\ \mathbb{R}^{n \times d}$ of a size compatible with the input vector $\boldsymbol{x}^{(t)}\ \boldsymbol{\mathcal{E}}\ \mathbb{R}^{d}$. The arranged cells hidden state is of dimension n, and so the gates contain compatible weight matrices $W_{fh}, W_{ih}, W_{oh}, W_h\ \mathcal{E}\ \mathbb{R}^{n\times n}$ and bias vectors $b_f, b_i, b_o, b\ \boldsymbol{\mathcal{E}}\ \mathbb{R}^{n}$. [30, p. 3]

A multilayer LSTM network

So far we have just considered a single layer n LSTM cells, called $\mathcal{L}_n$. In practice, several layers are frequently stacked to increase the complexity of the feature served by the network. At each time point t, the function $\mathcal{L}_n$ has two outputs $h^{(t)}, c^{(t)}\boldsymbol{\mathcal{E}}\ \pm\mathbb{U}^{\boldsymbol{n}}$. The hidden states $h^{(t)}$, it can be fed into the next layer, sequentially as seen in figure 7.

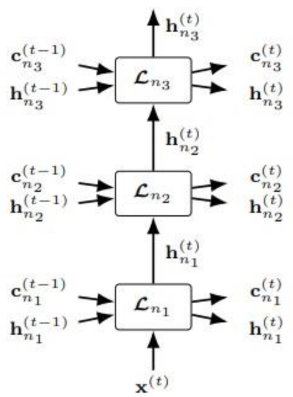


Figure 3-8 A graphical illustration of how the states pass through a multilayer LSTM [29]

A multilayer LSTM network of a function $\mathcal{S}$, where

$$\left(\boldsymbol{h}_{\boldsymbol{n3}}^{(t)}, \boldsymbol{c}_{\boldsymbol{n3}}^{(t)}\right) = \boldsymbol{\mathcal{L}}_{\boldsymbol{n3}}\left(\boldsymbol{h}_{\boldsymbol{n3}}^{(t-1)}, \boldsymbol{c}_{\boldsymbol{n3}}^{(t-1)}, \boldsymbol{h}_{\boldsymbol{n2}}^{(t-1)}\right),$$

$$\left(\boldsymbol{h}_{\boldsymbol{n2}}^{(t)}, \boldsymbol{c}_{\boldsymbol{n2}}^{(t)}\right) = \boldsymbol{\mathcal{L}}_{\boldsymbol{n2}}\left(\boldsymbol{h}_{\boldsymbol{n2}}^{(t-1)}, \boldsymbol{c}_{\boldsymbol{n2}}^{(t-1)}, \boldsymbol{h}_{\boldsymbol{n1}}^{(t-1)}\right)$$

$$\left(\boldsymbol{h}_{\boldsymbol{n1}}^{(t)}, \boldsymbol{c}_{\boldsymbol{n1}}^{(t)}\right) = \boldsymbol{\mathcal{L}}_{\boldsymbol{n1}}\left(\boldsymbol{h}_{\boldsymbol{n1}}^{(t-1)}, \boldsymbol{c}_{\boldsymbol{n1}}^{(t-1)}, \boldsymbol{x}^{(t)}\right)$$

can be represented by

$$\left(\boldsymbol{H}^{(t)}, \boldsymbol{C}^{(t)}\right) = \boldsymbol{\mathcal{S}}\left(\boldsymbol{H}^{(t-1)}, \boldsymbol{C}^{(t-1)}, \boldsymbol{x}^{(t)}\right),$$

The different number of cells $n_1, n_2, n_3, \dots\dots$, can be seen in each layer and so the hidden state and cell state vectors may be of different dimensions. The variables $\boldsymbol{H}^{(t)}$ and $\boldsymbol{C}^{(t)}$ Represent the sequence of all hidden states and cell states, respectively, at time point t. We note that in many applications, a network of stacked LSTM cells and building block for a larger model. For instance, in time series forecasting, an additional final layer is used to map the output from $\pm\mathbb{U}^n$, where n is the number of cells in the top layer, to time-series values in $\mathbb{R}$ [30, p. 3]

Feeding in Sequential data:

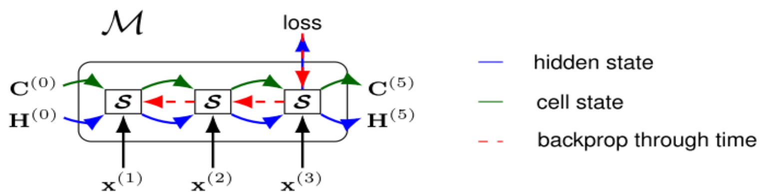


Figure 3-9 Feeding in Sequential data[29]

An example of the training test for an LSTM network with sequences of length $l = 3$. The function $\mathcal{M}$ takes in the initial states and the $input = x^{(1)}, x^{(2)}, x^{(3)}$. LSTM ignores the first two output of stacked LSTM. The path of the cell state is indicated by green lines, and the blue lines indicate the path of the hidden state. The red dotted lines display the route taken during the backpropagation to change the weights inside. $\mathcal{S}$. The weights inside $\mathcal{S}$ receive three additive updates. In time series forecasting, we need the model to have access to as many past observation as possible. Any memory of the time series before input, $input_i = (t_i + t_{i+1}, t_{i+2}, t_{i+3}, t_{i+4})$, must come from the cell state $\boldsymbol{C}^{(\mathbf{0})}$ and the hidden state $\boldsymbol{H}^{(\mathbf{0})}$. [30, p. 3]

3.4.2 Times series forecasting

After the recurrent model, say $\mathcal{F}$, has been trained, we would like to produce out-of-sample multi-step time series forecasts. In other words, given the time series $T = [t_1 , t_2, t_3,\ldots.., t_n]$, we would like to forecast k time points into the future to obtain $\hat{t}_{N+1}, \hat{t}_{N+2}, \ldots \ldots, \hat{t}_{N+k}$. [30, p. 3]

Iterated forecasting:

Train a ‘many-to-one’ function $\mathcal{F}$ such that

$$t_{i+l} \approx \mathcal{F}(\, t_i, \ldots \ldots \ldots t_{i+l-1})$$

For $i = 1, 2, \ldots\ldots, N - l$. The k-step forecast can be created by incrementally making one-step predictions using the previous forecast values, i.e.,

$$\hat{t}_{N+1} := \mathcal{F}(\, t_{N-l+1}, \dots \dots \dots t_{N-1}, t_N)$$

$$\hat{t}_{N+2} := \mathcal{F}(\, t_{N-l+2}, \dots \dots \dots t_N, t_{N+1})$$

$$\vdots$$

$$\hat{t}_{N+k} := \mathcal{F}(\, t_{N+k-l+1}, \dots \dots \dots t_{N+k-2}, t_{N+k-1})$$

We found that LSTM outperforms previous RNNs not only on regular language benchmarks (according to previous research) but also on CFL benchmarks. It learns faster and generalises better[31, p. 1339].

3.5 Related work

The forecasting of electricity demand has long been an area of research because of its importance to electricity providers. An accurate forecasting model allows an electricity provider to plan its generation and distribution more efficiently. This thesis conducted using power consumption for a household, which shares some of the characteristics of the problem this work considers.

Linton proposed kernel methods for predicting power consumption for several set of households using KNN machine learning models [12]. Unfortunately, this is not applicable in this thesis; we considered Neural networks for the prediction.

Ali, Mansoor et al. [32] proposed a method for short term (e.g. hourly) load forecasting at fine-scale (households). Our method uses hourly consumption data for a certain period (e.g. previous year) and predicts hourly loads for the next period (e.g. next six months).

Parate proposed a forecast model [17] ARMA, and ARIMA model to predict the forecast of an individual household and provided with comparative results.

Nogales et al. proposed an LSTM-based model to learn trends of electricity use for individual households as well as for various households[13], They have three approaches (naive, auto arima, LSTM model). They considered MAE to calculate the Error. The results shown that the LSTM approach is better than all, shows 19% better than auto arima.

Chakravorty et al. proposed Evolutionary Ensemble LSTM based Household Peak Demand Prediction [15]. They predicted for next minute peak demand. They are compared using two LSTM models, such as LSTM and EE LSTM. They found the results that EE LSTM performs both for cluster network or cluster network. However, this thesis only concerned with an hour and weekly prediction for a household.

4 SCIENTIFIC METHOD

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

4.1 Data collection and description

The thesis conducted uses a real data set containing electricity consumption minutely 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. [33]

Table 4-1 Variables in the dataset

date	Date in dd/mm/yy format
time	time hh:mm:ss in format
Global active power: household global minute-average active power	in kilowatt
Global reactive power: household global minute-average reactive power	in kilowatt
Voltage: minute-average voltage	in volt
global_intensity: household global minute-averaged current intensity	in ampere
Sub-metering 1: Average consisting mainly of a dishwasher, an oven and a microwave (hot plates are not electrical but are gas-powered)	in watt-hour of active energy
Sub-metering 2: This refers to the laundry room, which includes a washing machine, a tumble dryer, a refrigerator and a light.	in watt-hour of active energy
Sub-metering 3: It corresponds to an electric water heater and an air-conditioner.	in watt-hour of active energy

Dataset variable definitions:

Active power: The portion of power that, averaged over a full period of the AC waveform, lands in a net energy transfer in one direction is known to be active power. [34]

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

Voltage: Voltage is that the tension from the power source of the electrical circuit, which moves the charged electrons (current) into the conductive loop, allows them to undertake to work out how to illuminate a lightweight circuit.

Intensity: Intensity is that the speed, per unit time, at which a circuit transfers the current.

4.2 Data Preprocessing

Of all the steps involved in data processing, data preparation, while less troublesome, is one that takes more resources and more time to complete. The data collected is from a source have to be prepared for the data review process. The preparation of the data concerns the collection, cleaning, standardisation and transformation of the data into an optimised data set, that is to say, in a prepared format, usually tabular, suitable for the methods of analysis which have been designed during the design process. Several issues need to be avoided, such as invalid, undefined or incomplete values, repeated fields or out-of-range data. The dataset used in this thesis includes some missing values for the measurements (nearly 1,25% of the rows). All calendar timestamps are present in the data set. However, the values for some timestamps are lacking: the missing value is represented by the absence of a value between two consecutive semi-colon attribute separators. For example, the dataset showed missing values on Apr. 28, 2007.

4.2.1 Data Cleaning

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 dataset is shown above table 4-1, and it has nine attributes. To clean, the dataset, the data set is uploaded in Jupyter notebook(open-source web application which helps to create, share the codes, equations, visualisations and texts) [35] and imported data python libraries such as pandas, NumPy and matplotlib [35]. The index column of the dataset is set to $date\_time$ for further data preprocessing. In the table 4-2, it can be seen that the power consumed reading is started from 2006-12-16 (end of the December 2006) and, In the table 4-3, the last reading was on 2010-12-11 (till November 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.

Table 4-2 Dataset displaying the first five rows

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

Table 4-3 Dataset displaying the last five rows

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 to 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 looks fine without any 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 the descriptive statistics function. Pandas *describe*() is used to view some necessary statistical details like mean, count, percentile, standard deviation.

Table 4-4 Descriptive Statistics of the dataset

	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

Table 4-4 shows the statistics values of the dataset. In the next section, explained briefly about descriptive statistics.

4.3 Descriptive Statistics

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

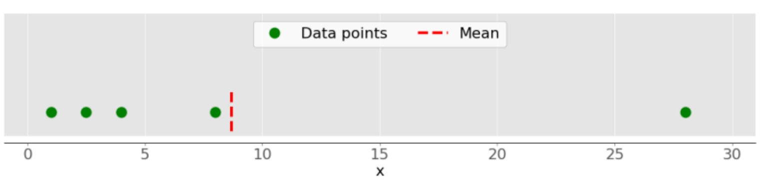


Figure 4-1 Mean [36]

Median: It is the middle element in sorted dataset.

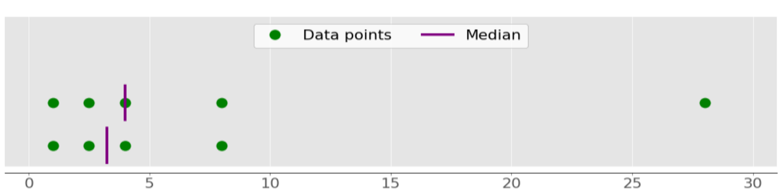


Figure 4-2 Median[36]

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

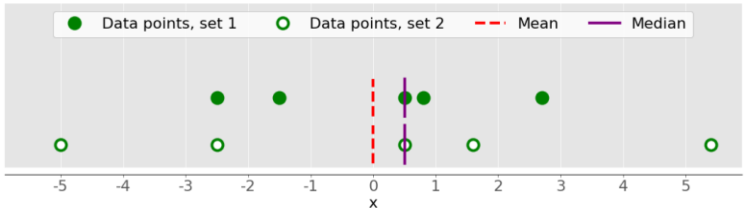


Figure 4-3 Variance [36]

Standard deviation: The standard deviation is another indicator of the distribution of results. It is related to the variance of the sample, as the standard deviation, s, The positive square root of the variance of the sample. The standard deviation is much more convenient than the variance since it has the same unit as the data point. [36]

Percentiles: The p percentile is the dataset element in such a way that p% of the dataset elements are less than value. In addition, $(100 - p)$ % of the elements are greater than or equal to that amount. If these of two elements in the data set, the percentile of the sample p is their arithmetic mean. Every dataset has three quartiles, and they are:

- The first quartile is the 25th percentile. It separates approximately 25% of the smallest objects from the majority of the data.
- The second quartile is the 50th percentile. About 25% of the products are Among the first and second quartiles but another 25% between the second and third quartiles.
- The third quartile is the 75th percentile sample. It separates approximately 25% of the most massive objects from the remainder of the dataset.

Ranges: It is the difference between the maximum and minimum element in the dataset.

4.3.1 Correlation between pairs of data:

To evaluate the relationship between the corresponding components of two variables in the data collection. The below figure shows the correlation between the two data[36]

- A positive correlation: it exists when larger values of x correspond to larger values of y and vice versa.

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

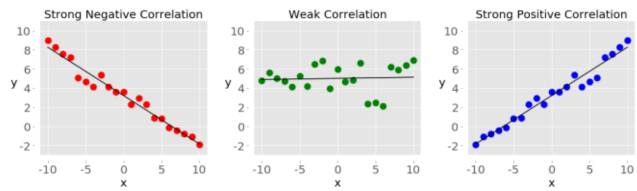


Figure 4-4 Measures of Correlation

The correlation between the attributes in the dataset considered in the thesis is visualized to know if their any relation exits between the attributes. The comments are mentioned below.

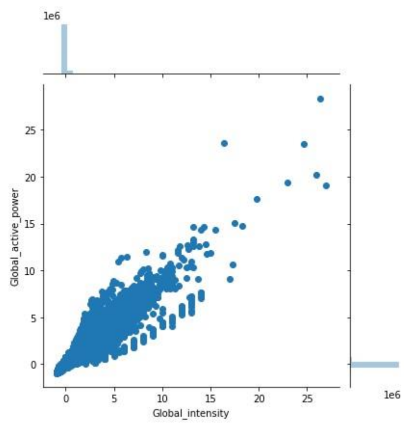


Figure 4-5 Correlation between active power and intensity

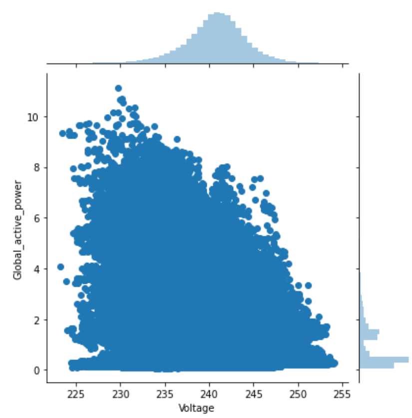


Figure 4-6 Correlation between Global active power and voltage

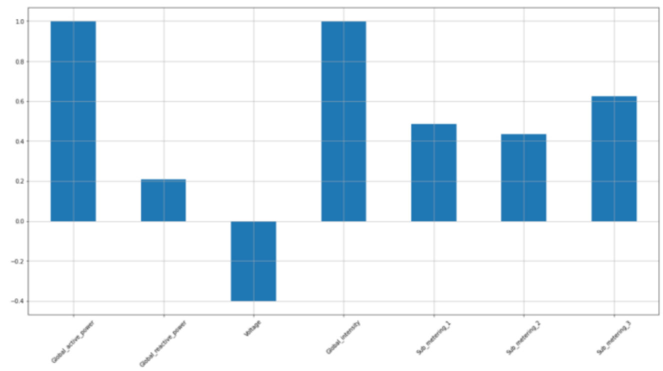


Figure 4-7 Comparison of Correlation of all attributes

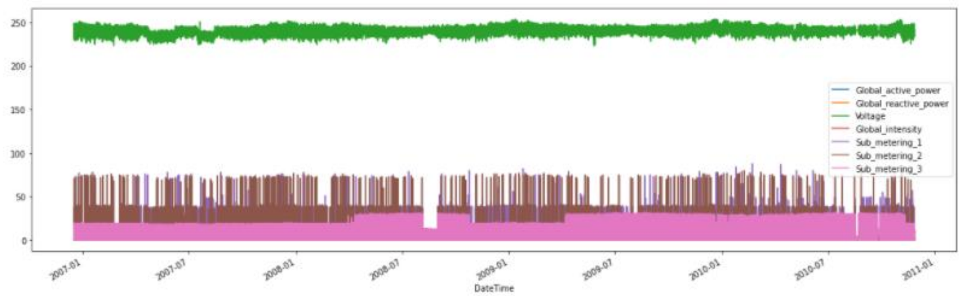


Figure 4-8 Comparison of Correlation

Figure 4-5 shows the positive correlation between global active power and intensity. In fig 4-6 shows the negative correlation between global active power and the voltage. In figure 4-7, we can see that every attribute showing the relation but voltage shows the negative correlation and the same as plotted with a time-series graph. It is essential to know the correlation between the variables because when it is correlated one variable changes in value, the other values tend to change in the specific direction. Correlation is the most crucial point to know before making any prediction.

4.3.2 Covariance

It is a metric that quantifies the intensity and direction of the relationship between the two variables:

• When the correlation is positive, the covariance is positive as well. A stronger relationship leads to a higher covariance value. It shows a close association with global active power and global strength, as seen in figure 4-5.

• When the correlation is negative, the covariance is harmful, too. A better relationship leads to a lower (or higher) relationship. It shows a negative covariance between global active power and voltage, as seen in figure 4-6.

4.3.3 Correlation Coefficient

The symbol r indicates the correlation coefficient. The coefficient is a further measure of the correlation between data. [36]

- The value of r > 0 implies a positive correlation.
- The value of r < 0 implies a negative correlation.
- The value of r = 1 is the highest possible value of r, which refers to the perfect positive linear relationship between the variables.
- The value of r = −1 is the lowest possible value of r, which refers to the ideal negative linear relationship between the variables.
- The value of r is 0, or when r is about zero, implies that the association between variables is small.

4.3.4 Visualising Data

We visualized the dataset to check the correlation between the variables, which gives an idea to select a feature for our prediction. To present the data visually using the python library, Matplotlib is a very convenient and commonly used library for the visualisation of any datasets. In figure 4-11 its shows the correlation between each variable, if it has positively correlated, it shows 1 and least as -1.

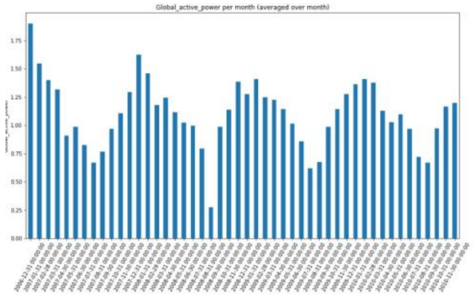


Figure 4-10 Histogram

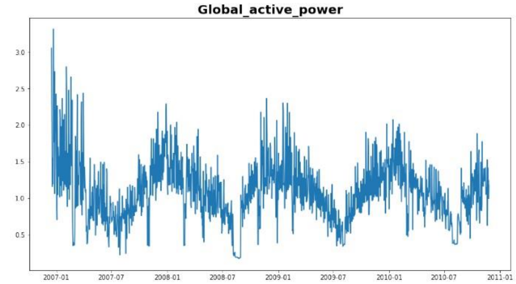


Figure 4-9 X- Y plot

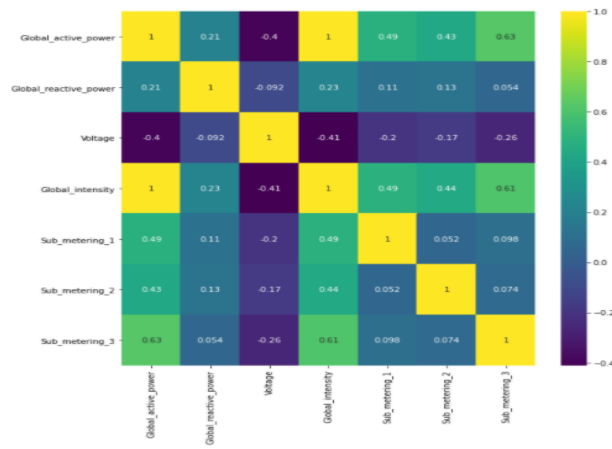


Figure 4-11 Heat maps

4.4 Exploratory data analysis

This section contains some exploratory analysis of the data set used in this thesis and highlights the crucial characteristics found. Data exploration consists of a preliminary review of the data, which is essential for understanding the type of information that has been collected and what it means. In conjunction with the knowledge acquired during the description issue, this categorisation can decide which method of data analysis is ideally suited to the definition of the model. In the below figure 4-12 and figure 4-13, We try to observe the components of the time series data, which are trends, seasonality, and residue. As in figure 4-12, we do not see any trends like positive or negative, and also do not see seasonality, residue. The power flow is showing oscillation with the same frequency; hence it is a stationary dataset so we can make use of this model to predict promising prediction. The same characteristics observed in figure 4-13, the power consumption happened in 4 years. We considered global active power for the prediction because it is correlated to every variable, and it is an active load. Further sections, we look into deeper to visualise the power consumption on several periods.

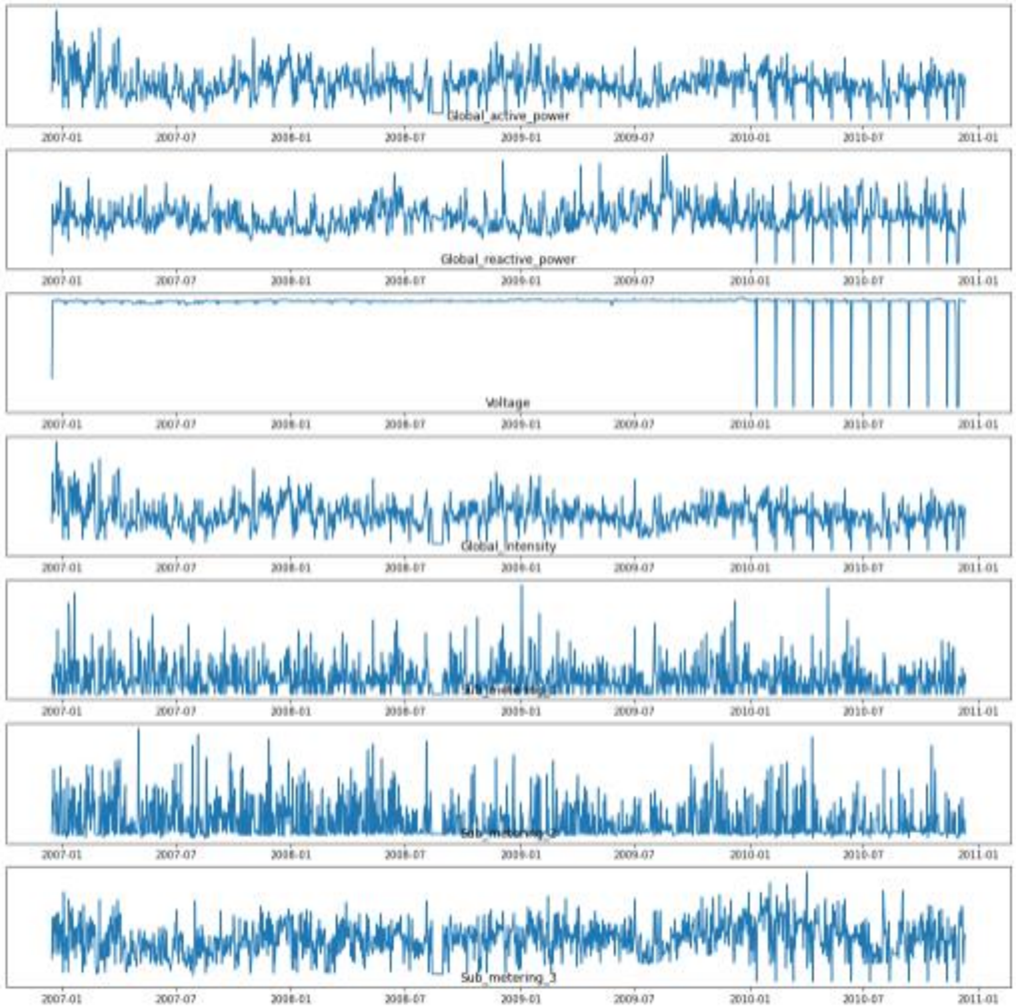


Figure 4-12 Visualisation for all attributes for four years

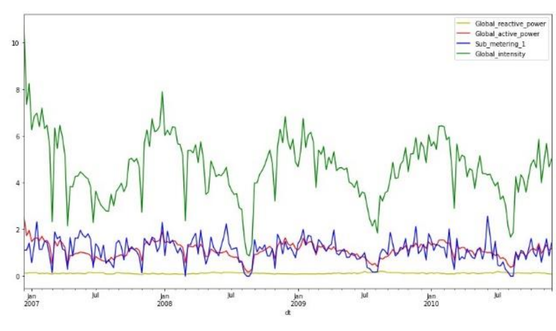


Figure 4-13 Visualisation for a week resampled for all the attributes for four years

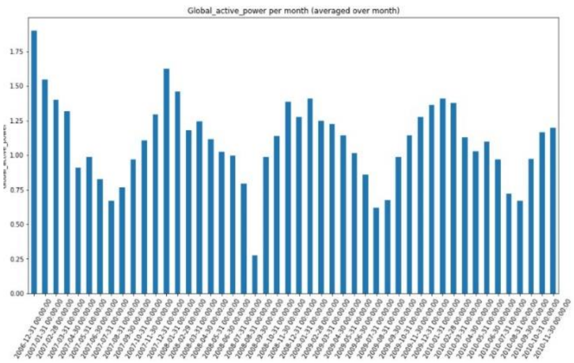


Figure 4-14 Visualisation for Global active power resampled over a month for four years

The visualisation 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 are correlated to it. The six visualisations made is shown below in the table. Data Visualization for periods is also one of the goals of the thesis to analyse power consumption in a period.

Table 4-5 Average power consumption several during periods

	Power consumption during several periods	Observations
1	Average power consumption for each year	The year 2006 is more consumed than the other three years is observed.
2	Average power consumption for each month	Power consumed is more in December and January, then from January it continually decreases and constantly rises till December
3	Average power consumption for each day in a month	Power consumed over each day is constant, but during the month-end, it is high.
4	Average power consumption for each day in a week	Power consumed more during the weekends than weekdays is observed.
5	Average power consumption for each hour in a day	Household consumed maximum power during 21:00 hour and half of its during 10:00 hour

6	Average power consumption for each month in 4 years	Power consumed is more in December and January, then from January it continuously decreases and always rises till December

We can say that from the observations, the household consumed more power during the winter season, than the summer season. The power consumption observed more during the weekend than weekdays, in the time 21:00 hr, in the night time. Moreover, in the morning because generally, they use the shower and kitchen. The august month they have consumed the least

4.4.1 Average power consumption for each year in 4 years

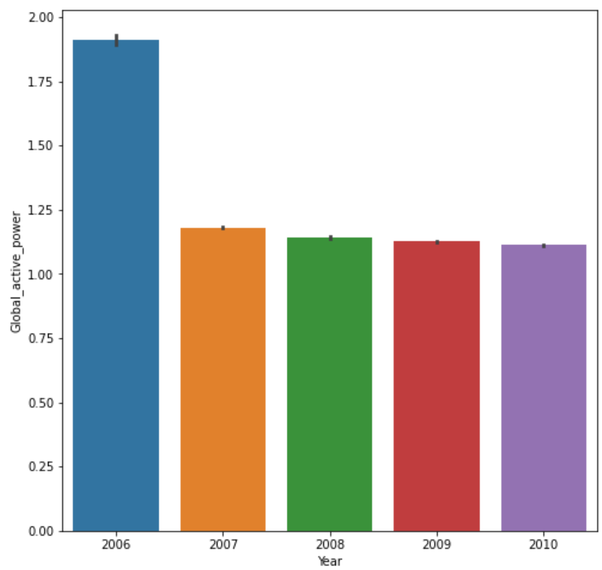


Figure 4-15 Average power consumption for each day in 4 years

The above graph shows the average power consumption for each year in for four years. The horizontal axis depicts four years starting from 2006 to 2010 through which the energy consumed was observed, and the vertical axis depicts the number of a kilowatt of energy consumed over time. From the bar graph, we can tell that there is a power demand can be observed in 2006. The minimum power consumed in the household is in the year 2010, which is showed in figure 4-15.

4.4.2 Average power consumption for each month

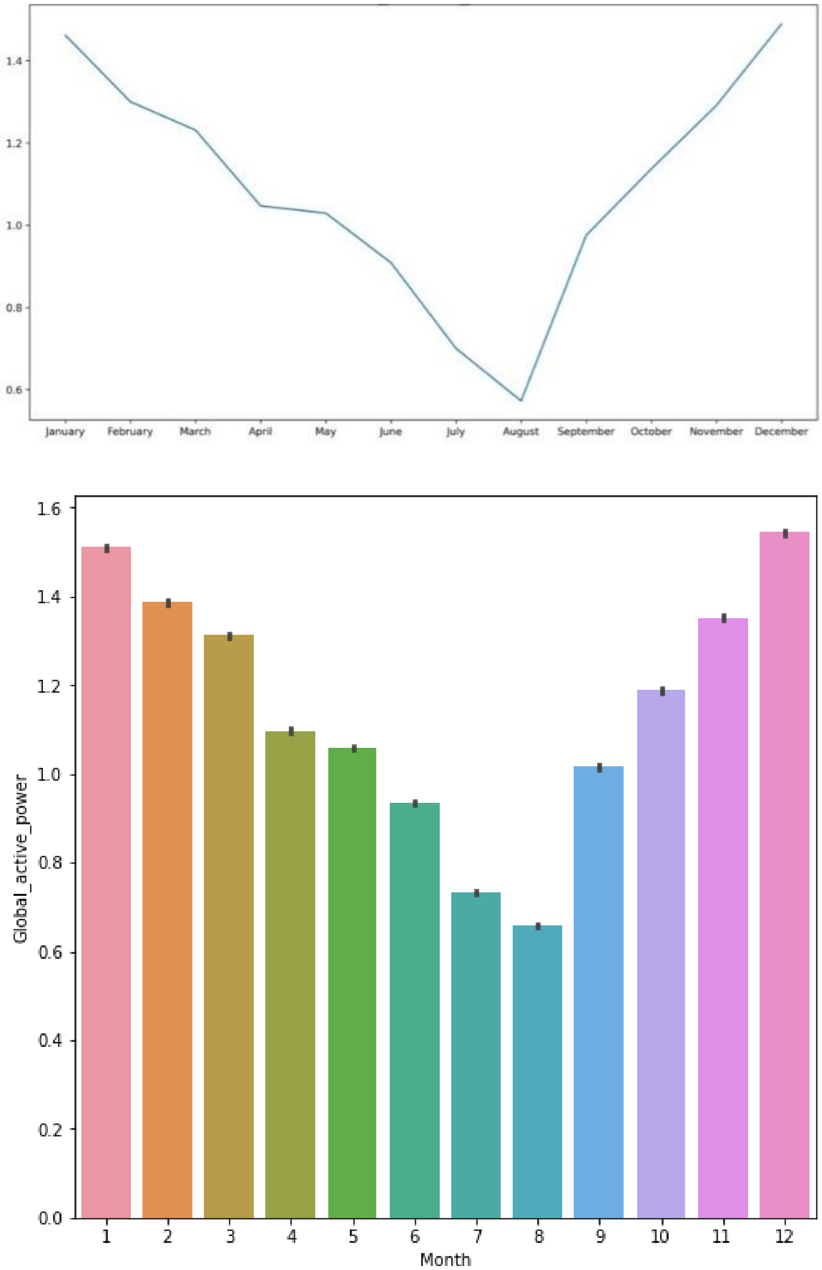
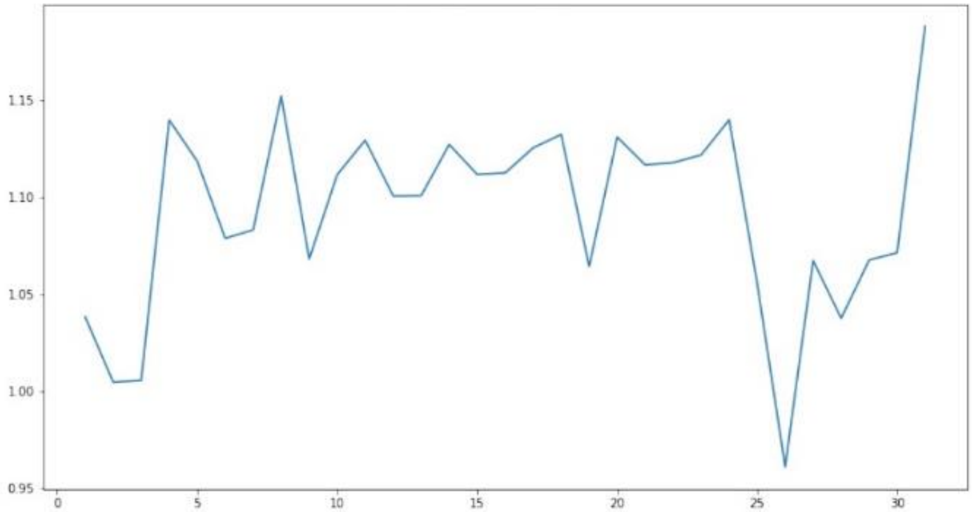


Figure 4-16 Average power consumption for each month

The above graph shows the average power consumption for each month in for four years. The horizontal axis depicts months starting from January until December, and the vertical axis depicts the number of a kilowatt of energy consumed over time. From the bar graph, we can tell that there is a power demand can be observed in January and in December. The minimum power consumed in the household is in August year, and there is a rise in the power consumption starting from

September till the end of January. Then there is a fall in power consumption starting from January until the end of August. Whereas with the maximum power consumed, there are variations observed and the highest power consumed is shown in the month and January and December.

4.4.3 Average power consumption for each day in a month



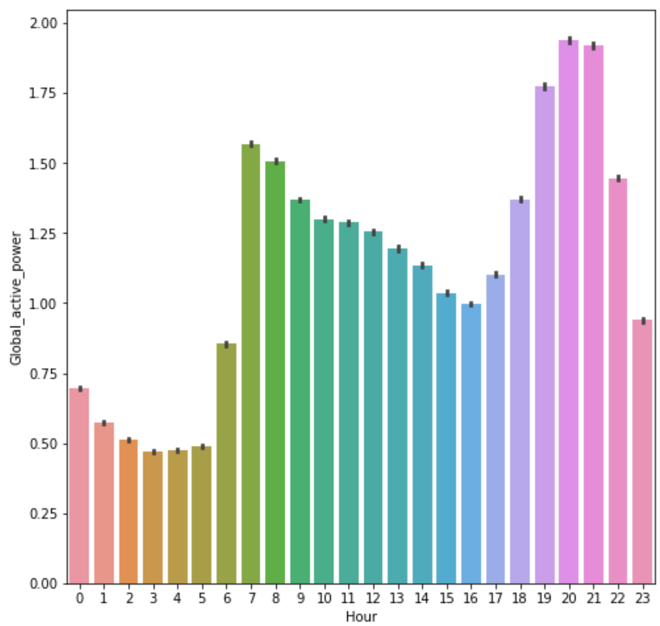


Figure 4-17 Average power consumption for each day in a month

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 Power consumed over the month.

4.4.4 Average power consumption for each day in a week

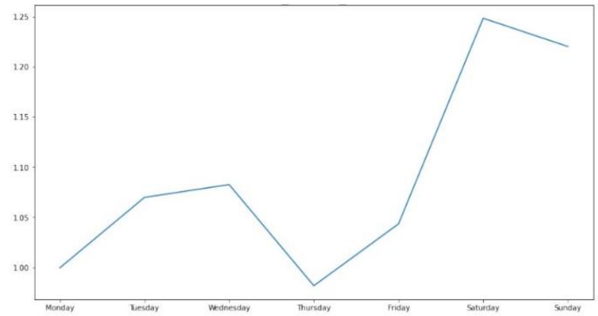


Figure 4-18 Average power consumption for each day in a week

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 power consumed over the week. From the line graph, we can tell that there is a power consumed can be observed more on weekends than weekdays. The average minimum power consumed in the household was on Mondays and Thursdays. Then there is a rise in power consumption starting from Friday until the end of Saturday. Whereas with the maximum power consumed, there are variations observed and the highest power consumed is shown on Saturday.

4.4.5 Average power consumption for each hour in a day

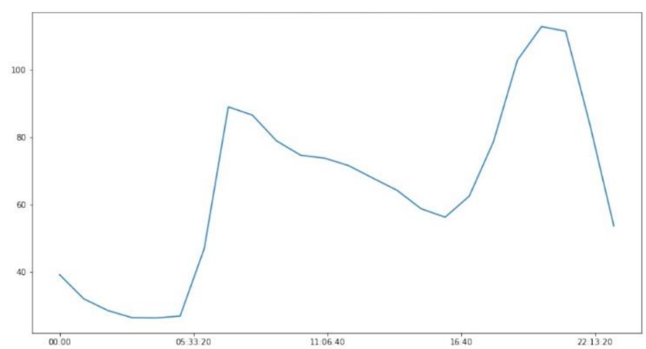


Figure 4-19 Average power consumption for each hour in a day

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 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 a 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.

4.4.6 Average power consumption for each month in 4 years

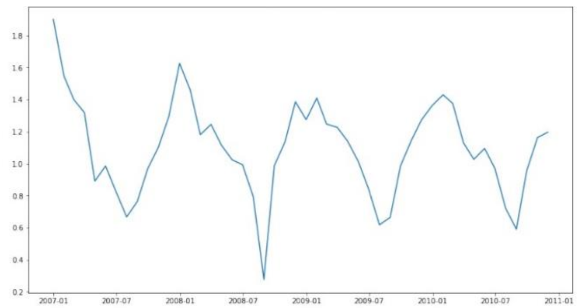


Figure 4-20 Average power consumption for each month in 4 years

The above graph shows the average power consumption for month day in for four years. The horizontal axis depicts four years starting from 01/2007 to 12/2010 through which the energy consumed was observed, and the vertical axis depicts the number of a kilowatt of energy consumed over the time. From the line graph, we can tell that there is a power demand can be observed in January and in December. The minimum power consumed in the household is in August of every year, and there is a rise in the power consumption starting from September till the end of January. Then there is a fall in power consumption starting from January until the end of September. Whereas with the maximum power consumed there are variations observed and the highest power consumed is shown in the year 2006, and the lowest power consumed is shown in the year 2010.

4.4.7 Power consumptions in seasons :

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

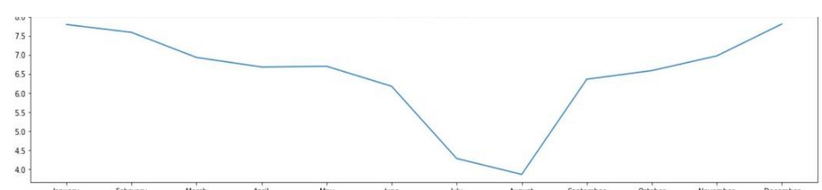


Figure 4-21 Power consumptions over seasons

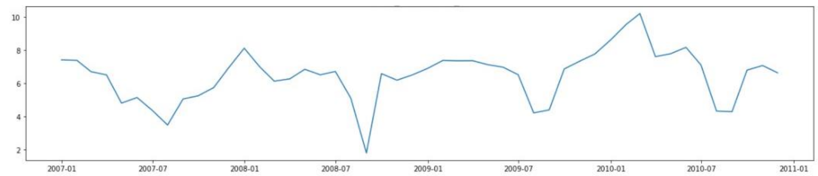


Figure 4-22 Power consumptions in seasons for 4 years

Power demand is less during the summer due to summer holidays, even for industrial sectors also it applies the same in August, for example[37]. The above line graph depicts the average weekly demand at reference temperatures for a typical July to June period for the residential sector.

4.5 LSTM Algorithm for Time series prediction:

The step by step approach for building the LSTM model for time series prediction is explained in the below section

4.5.1 LSTM Algorithm

Problem statement: Predict the power for next hour based on past 24 hours records from a dataset of 4 years. The several steps involved have explained below:

Resampling: The dataset collected from the open-source had minutely recorded values; we had resampled it to daily readings according to the problem.

Separation: The data set is separated into two sets, training set and testing sets. The training set for training the model to learn from its values and testing sets are used to plot against the trained set. In the thesis, we considered training set as from the year 2006 to 2009 and for the testing set the year 2010.

Model Training:

Data Preprocessing: We divided the whole training set and testing sets for the standard weeks according to the problem. So for training 159 weeks, testing set 46 weeks.

Training set: The first row represents time stamps weekly readings and The second row represents some Global active power values at that time. For training a model, we are assigning timestamps to learn the values.

t_1	t_2	t_3	t_4	t_5	t_6	………	t_{158}	t_{159}
x_1	x_2	x_3	x_4	x_5	x_6	………	x_{158}	x_{159}

For an explanation of how it works, We had considered 3-time steps. We are training the model for three timestamps and creating our first record, which is already are there in the dataset. The data set is trained for 3 time steps, and for each time steps, we are assigning the next step values.

X Y

t_1	t_2	t_3	
x_1	x_2	x_3	x_4
x_3	x_4	x_5	x_6
x_5	x_6	x_7	x_8
…….	…….	…..	………………..
x_{156}	x_{157}	x_{158}	x_{159}

The shape of the training set is (159, 1). We have to train the model for the particular stamp now so we are converting 2D array into 3D arrays to store the values. (159, 1, 1). So We have assigned only the row already and iterating allover the length dataset.; Then we are appending the output value to the next row. We build a model of LSTM with layers and optimisers to calculate the values.

X Y

	t_1	t_2	t_3	
1 step	x_1(assigned already)	x_2(assigned already)	x_3(assigned already)	x_4(assigned already)
2 step	x_2	x_3	x_4(assigned is taken as input)	x_5(calculated output from new input)
3 step	x_3	x_4(assigned is taken as input)	x_5(calculated output from new input)	x_6(calculated output from new input)
4 step	x_4(assigned is taken as input)	x_5(calculated output from new input)	x_6(calculated output from new input)	x_7(calculated output from new input)
	-------	-------------	– – – – – –	– – – – – – – – – – – – –
156 step	x_{156}(calculated output from new input)	x_{157}(calculated output from new input)	x_{158}(calculated output from new input)	x_{159}(calculated output from new input)
157 step	x_{157}(calculated output from new input)	x_{158}(calculated output from new input)	x_{159}(calculated output from new input)	x_{160}(calculated output from new input, (we predicted the value from its data)

After the 160th week, The same process can be continued. However, it depends on the component values and also timestamps we give to the LSTM model, This depends on the Error, So we have to check frequently so that the model is learning correctly or not and change according to that.

Testing set :

t_1	t_2	t_3	t_4	t_5	t_6	………	……..	t_{47}	t_{48}
x_1	x_1	x_1	x_1	x_1	x_1	………	……..	x_1	?

Model evaluation:

We got the values from the model; now, we can evaluate with the test data. The error values are Mean squared Error, and Root means square to know the Error between the curves. It shows the actual difference between the learning curve(predicted) and the actual one.

Plotting :

The plot against trained values and test values to know how predicted set follows the actual test set, Which means we are plotting Actual of 2010 against Predicted one.

The experiments performed explained in the next chapter.

5 EXPERIMENTAL SETUP

This chapter introduces the dataset used for the prediction task and reports the outline and results of the experiments conducted. A specific focus is given to the evaluation of results, which compared with each other. The proposed LSTM model was implemented and executed on Deep learning library, Keras (version 2.4.3) in Programming language, Python (version 3.8.3) with python libraries such as Pandas(version 1.0.5), Numpy(version 1.18.5), for data visualization Matplotlib(version 3.2.2) and Machine learning libraries such as Scipy(version 1.5.0), Statsmodels(version 0.11.1), and Sklearn(version 0.23.1) in Anaconda software(version 4.9.1). The analysis runs on Intel® core™ i3-6006U processor running at 2.00GHz 4.00 GB of RAM, running windows version 10.

For forecasting the household electricity consumption in France, we have collected real data of approximately four years, i.e. from December 2006 to November 2010 from the open-source platform [33]. The attributes of the data frame are date-time, Global active power, Global reactive power, voltage, Global Intensity, Submetering 1, sub-metering 2, sub-metering 3. The dataset is divide into two separate datasets. The training data as (data from December 2006 to December 2009) and test data (January 2010 to November 2010) shown in figure 1 below.

On the other hand, the second data set created as per standard weeks and separating them into two groups: one for training purposes (Only data of 159 weeks, starting from December 17, 2006(first Sunday) till the end of the week in 2009) and another for performance testing purposes (Only data of 46 weeks, Starting from January 3, 2010(first Sunday) till the end of the week in 2010 shown in figure 2 below.

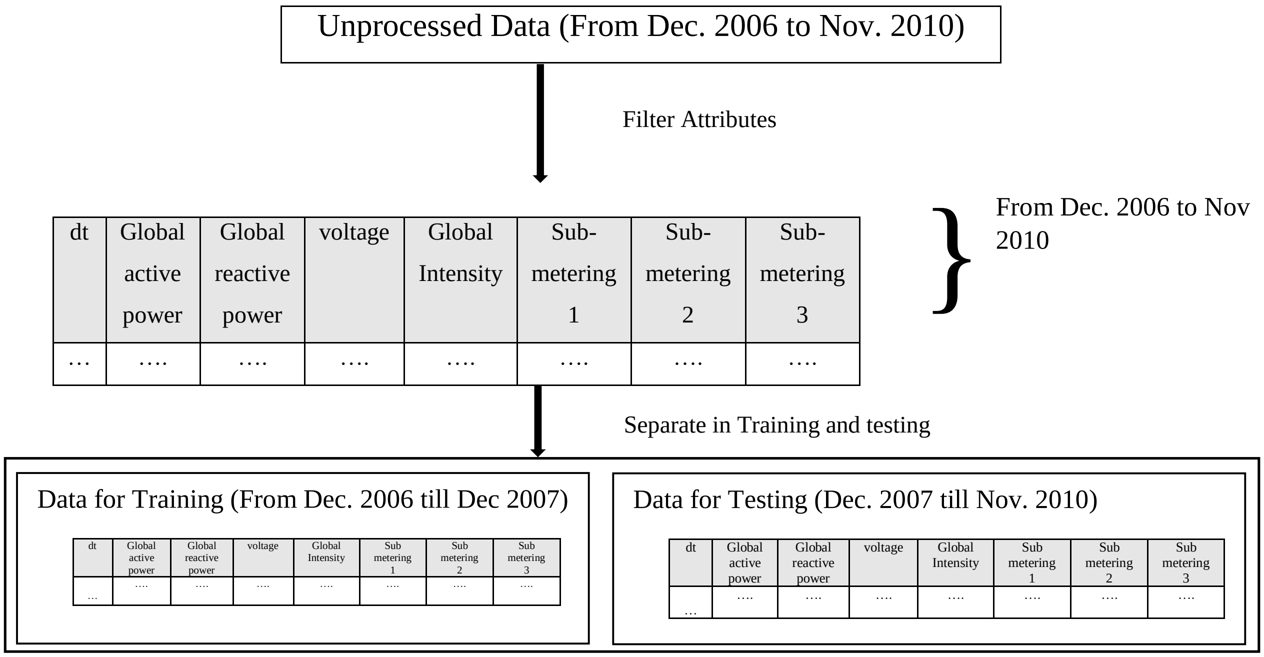


Figure 5-1 Schematic overview for An hour prediction

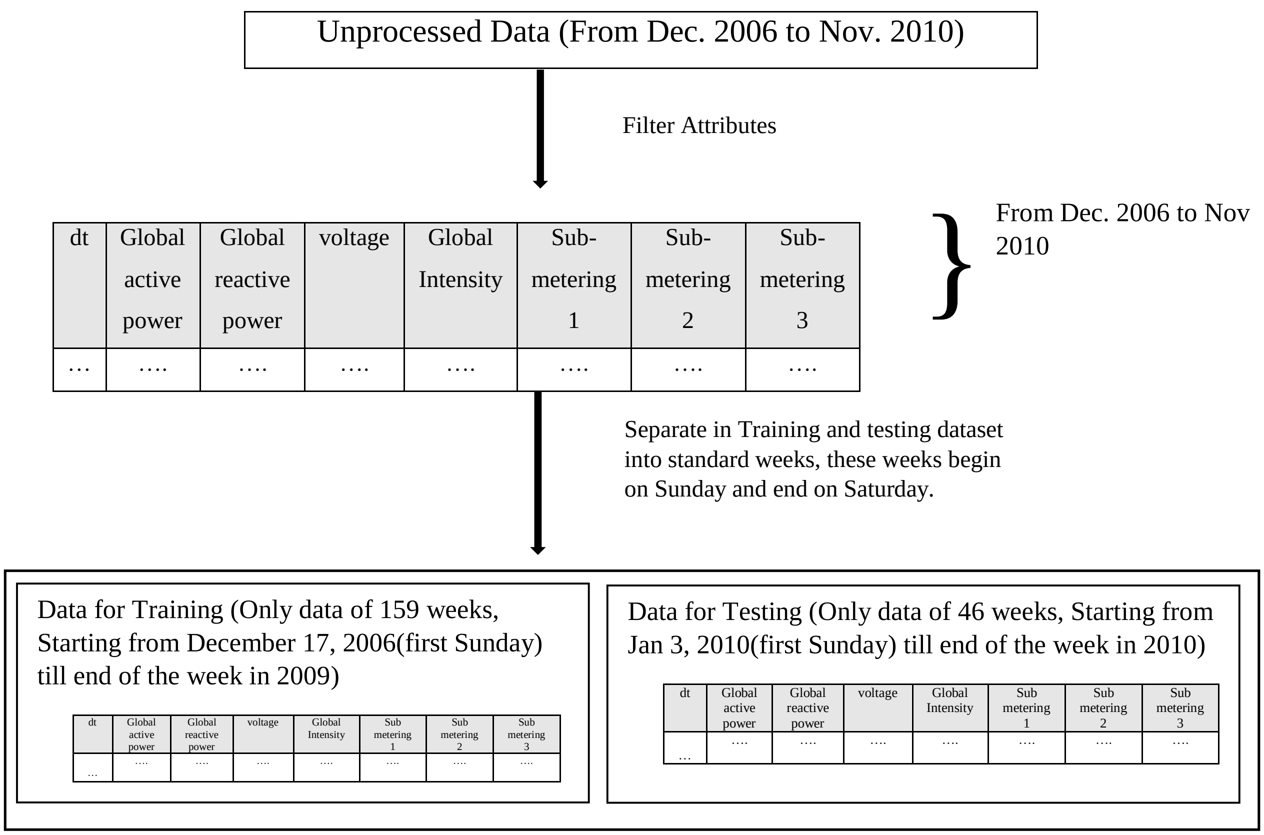


Figure 5-2 Schematic overview for a week prediction

5.1 Prediction for an hour

Preparing the LSTM Model for an hour power consumption prediction, the following steps involved, as shown below:

After the data processing and data cleaning, the dataset is ready to make the LSTM model. At first, the data set is framed as a supervised learning problem and normalising the input variables. Framing the supervised learning problem as predicting the power consumption at the current hour (t). Predicting the power consumption for the next hour based on the power consumption over the last 24hrs [38]

The Next step is to define and fit the LSTM Model. First, Splitting the data sets into the train and test data sets $(train, test)$ as shown in fig 5-1. To speed up the training model, fit the model on the first year of data, then evaluate the remaining three years of data. Then split the $(train, test)$ into input and output as $(train_x, train_y)$ and $(test_x, test_y)$. Now the $(train_x, train_y)$ and $(test_x, test_y)$ are in 2D format. The LSTM expects the data has to be in 3D format. So, convert $(train_x, train_y)$ and $(test_x, test_y)$ into the 3D format by reshaping function.

The next step is to fit the LSTM model, So, define the model as sequential and layers as 100 and the output layer as 1. The input shape will be a 1-time step with eight features—the Mean Absolute Error (MAE) loss function and the optimiser adam as stochastic gradient descent. The model is fit for 20 training epochs with a batch size of 70. To track the training and test loss during training, added the validation data argument in the $fit()$ function, which is already explained in section 2.

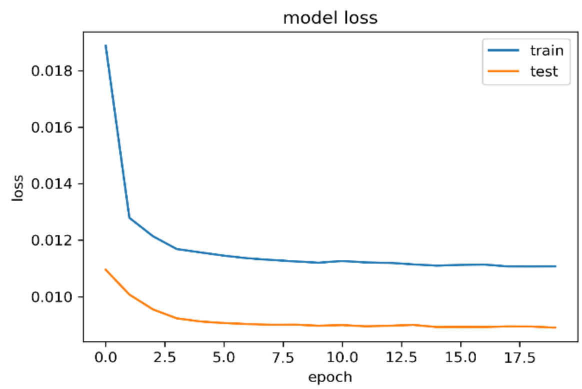
The next step is to evaluate the Error, which is defined below. [39, p. 9]

Mean Squared Error - Average squared difference between the actual and predicted value.

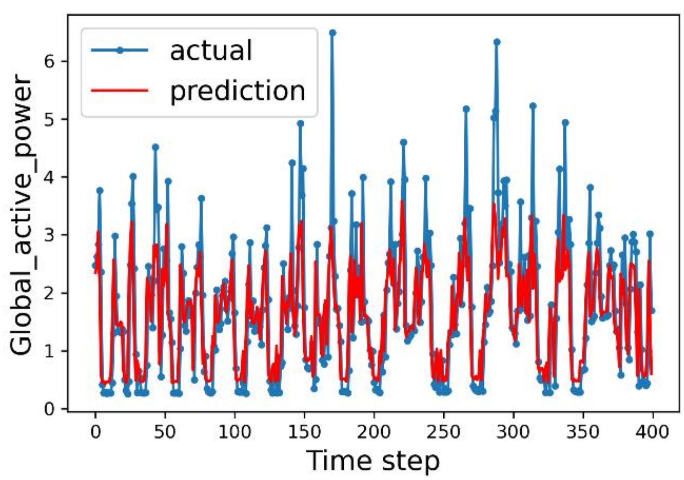
$$MSE = \sum_{n=1}^{k} \frac{(Actual - Predicted)^2}{k}$$

$$RMSE = \sum_{n=1}^{k} \sqrt{\frac{(Actual - Predicted)^2}{k}}$$

RMSE found to be, $RMSE = 0.607$ in our experiment. The prediction model is good. The RMSE is the square Root of variance; it indicates the closeness data points of the model. Lower the RMSE gives a better fit as the standard deviation of the unexplained variance and has the benefit of being in the same units as the answer variable that is the primary purpose of the model is prediction. [40]



The next step is to plot the actual and the predicted power consumption. From the graph, the predicted curve is following the actual one. The LSTM model predicted the next 400 time stamps (1-time stamp means one hour) which means the LSTM model learned the data and predicted well.



5.2 Prediction for a week

Preparing the LSTM Model for a week prediction power consumption, the steps following involved, as shown below:

After the data processing and data cleaning, the dataset is ready to make the LSTM model. At first, the data set is framed as a supervised learning problem and normalising the input variables.

Framing the supervised learning problem as predicting the power consumption at the current hour (t). Predicting the power consumption for the next week based on the power consumption over the last week (7 days), last two weeks (14 days), last three weeks(21 days), and the previous four weeks (28 days).

The Next step is to define and fit the LSTM Model. First, Splitting the data sets into the standard weeks as train and test data sets $(train, test)$ as shown in figure 5-2. For the training set (Only data of 159 weeks, starting from December 17, 2006(first Sunday) till the end of the week in 2009) and another for performance testing purposes (Only data of 46 weeks, Starting from January 3, 2010(first Sunday) till the end of the week in 2010). Now the data set of $(train, test)$ match the specific dates that defined as the boundary on the standard weeks for each data set.

This is where the model is used to make a 1-week forecast, so the detailed data for that week is included in the model, so it is also used because of the basis for the next week's forecast. This can be both realistic in terms of how the model might also be used in practice and useful to the models, allowing them to make use of the most superficial available data. The concept is seen in the following.

Input,	Predict
[week 1]	week 2
[week 1+ week 2]	week 3
[week 1+ week 2 + week 3]	week 4

[week n-1 + week n-2 + week n-3]	week n (present week)
[--------- + ------------- + week n]	week n+1 (next week)

Then split the $(train, test)$ into input and output as $(train_x, train_y)$ and $(test_x, test_y)$. Now the $(train_x, train_y)$ and $(test_x, test_y)$ are in 2D format. The LSTM expects the data has to be in 3D format. So, convert $(train_x, train_y)$ and $(test_x, test_y)$ into the 3D format by reshaping function.

The next step is to fit LSTM model, So, define the LSTM 200 neurons in the first hidden layer and one neuron in the output layer for predicting the power consumption for each day in a week. The input shape is unrelated when giving an input. The LSTM connected with 200 neurons will interpret the features learned by LSTM layer [41]. To evaluate Error, Mean Absolute error(MAE) loss function and the optimiser adam as stochastic gradient descent. The model is fit for 70 training epochs with a batch size of 16.

After the above step, we have the data of the prior week to predict the next week. So, the prior week data is collected in an array of standard weeks. The data will be in the form of regular weeks. So, to predict next week, we need to retrieve the data of last days information for this, we need to use the flatten data function to break weekly readings into individual days reading which gives us eight parallel time series data.

The next is to obtain the last seven day's power consumption (feature index 0). Then we need to parameterise this as we did for the training data so the number of previous days used as input and then make a prediction using the fit model and therefore the input file and retrieve the vector of seven days of output. The function implements this and to take the model to fit in the training dataset. The history is already stored in the model, and it expects the input time steps. Then the model fits and evaluate the model. It prints the overall RMSE across the number of days given in the input and the per-day RMSE for each lead time. [42]

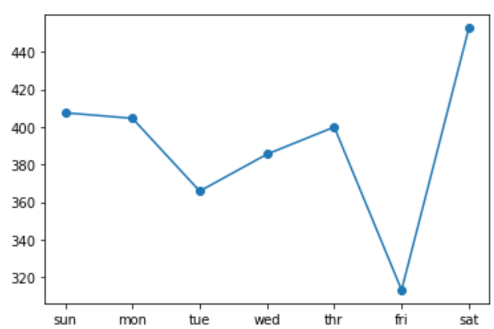


Figure 5-4 Forecast for a week by taking the past 7 days values

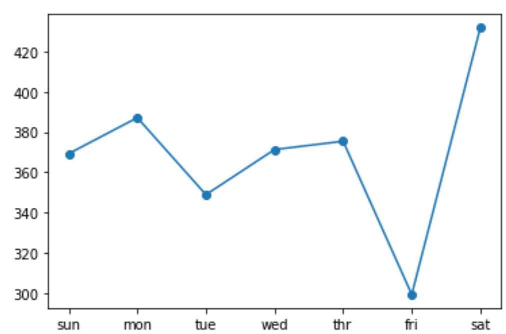


Figure 5-3 Forecast for a week by taking the past 14 days values

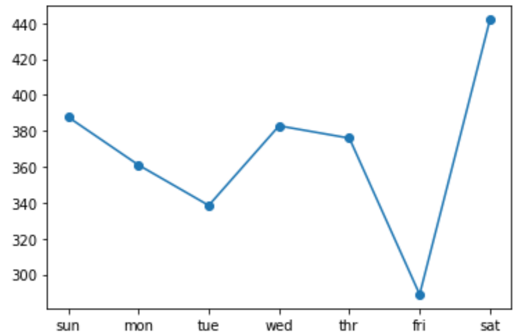


Figure 5-6 Forecast for a week by taking the past 21 days values

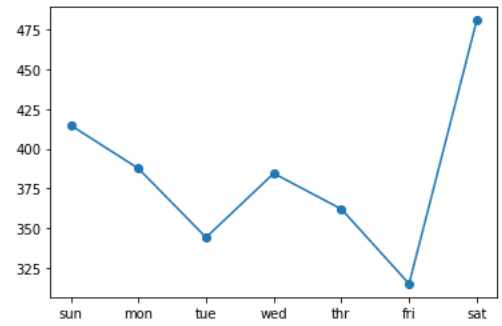


Figure 5-5 Forecast for a week by taking the past 28 days values

The plot shows that Tuesdays and Fridays are more comfortable with making forecast than the end day of the standard week, Saturdays (comparing to the overall consumption Saturdays shows more difference than weekdays).

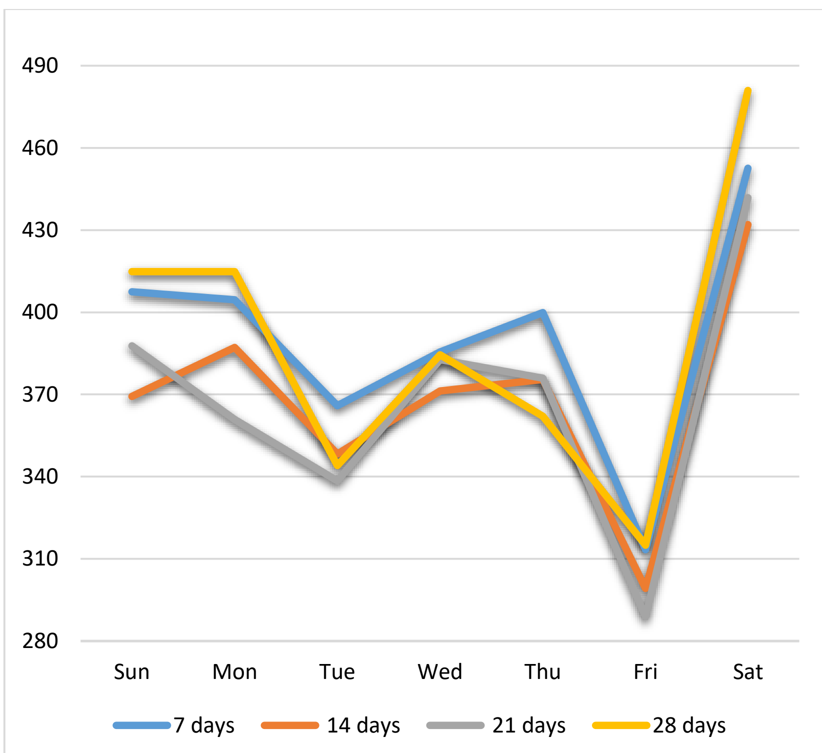


Figure 5-7 Prediction for overall past days

5.3 Result & Discussions of the analysis:

Analysis 1: The power prediction for an hour

The analysis performed on a household dataset to predict the power consumption based on the preceding 24-hour recording. The household dataset has four years of minutely recorded power consumption readings. The main goal is to build an LSTM model to predict the power consumption for the next hour and to evaluate the performance of the model. The main findings are the Mean absolute Error, Mean squared Error, and Root Mean squared Error. This error demonstrates the error values between the actual and the predicted power consumption values. In table 5-1 and table 5-2 below, the calculated error values and actual and predicted power consumption is shown respectively. The results show that the predicted model is approximate, because the RMSE value found is lower than the mean of standard deviation.

Table 5-1 Error Values for an hour Prediction

Mean Absolute Error	0.4360
Mean Squared Error	0.3687
Root Mean Squared Error	0.6072

Table 5-2 Error-values Actual vs Predicted

Test data (From Dec 2007 to end of Dec 2010 in hours)	Actual (kilowatt)	Predicted (kilowatt)
0	2.4749	2.3105
1	2.6266	2.4274
2	2.8342	3.0477
3	3.76716	2.1767
:	:	:
25813	1.7533	1.1679
25814	1.1149	0.4766
25815	0.3205	0.7653

Analysis 2: The power prediction for a week

The analysis performed on a household dataset to predict the power consumption based on the preceding 7, 14, 21, 28 days recordings. The household dataset has four years of minutely recorded power consumption readings. The main goal is to build an LSTM model to predict the power consumption for the next week and to evaluate the performance of the model. The main findings, Root Mean squared Error for the next seven days. This error demonstrates the error values between the actual and the predicted power consumption values. In table 5-3 below, the calculated error shown. The results show that the RMSE for the past 21 days is lower compared to the 7, 14, 28 days. The overall average predicted forecast for the next week is about 380.19 kilowatt.

The LSTM model performed well to predict on the weekdays like Tuesday and Friday because the lowest RMSE difference can be seen with the RMSE of overall power consumption. The prediction for Saturdays is having a higher difference in the overall RMSE when compared to the Saturdays. So from the analysis, It is hard to predict the accurate prediction on Saturdays. The results are shown in the below table 5-3.

Table 5-3 Prediction for a week

LSTM Model Prediction	Predicted (Kilowatt) 7 days	Predicted (Kilowatt) 14 days	Predicted (Kilowatt) 21 days	Predicted (Kilowatt) 28 days
Overall power consumption for next week	391.92	370.91	370.62	387.33
Sunday	407.5	369.3	387.7	414.8
Monday	404.6	387.2	360.9	387.6
Tuesday	365.9	348.3	338.4	344.1
Wednesday	385.5	371.3	382.7	384.5
Thursday	399.9	375.5	375.9	362.0
Friday	313.3	299.2	288.8	314.9
Saturday	452.6	432.0	441.9	481.0

6 CONCLUSIONS

The thesis aimed to predict the power consumption of a household from its past year recorded power consumption values and to perform data analysis on the household data set, with the implementation of deep neural networks, LSTM model. The data analysis provided the results of the power consumed in the household at different times are, the minimum power consumed in the household in August of each year and the maximum power consumed in December and January. The highest power consumed in 2006, and the lowest in 2010. The actual power consumption is more in between 20:00 hr and 21:00 hr. Power usage higher on weekends than on weekdays. The power consumption is more in winter because of the usage of heaters compared to the summer. The forecast for an hour provided the results with an RMSE error of 0.607 kilowatts. Which suits well, the prediction is consistent. The week forecast is shown that the average RMSE of overall power consumption was 380.19 kilowatts and results show that accuracy comes only for Tuesdays and Fridays compared to Saturdays.

6.1 Future works

This thesis considered only one household due to different characteristics of households, and it would be interesting to consider the effects that set containing several households on the algorithms. Further, weather data could also be introduced to see its impact.

The implementation of the LSTM model in this thesis used an iterative method to achieve a multi-step prediction. Several alternatives to the iterative method have been proposed that reduce the effect of errors aggregating over each step. The direct method could be implemented, which may perform better.

The statistical methods such as ARMA, ARIMA, CNN, LSTM-CNN are suitable to explore.

REFERENCES

[1] E. E. Michaelides, *Alternative Energy Sources*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[2] F. U. M. Ullah, A. Ullah, I. U. Haq, S. Rho, and S. W. Baik, “Short-Term Prediction of Residential Power Energy Consumption via CNN and Multi-Layer Bi-Directional LSTM Networks,” *IEEE Access*, vol. 8, pp. 123369–123380, 2020, doi: 10.1109/ACCESS.2019.2963045.

[3] G. Zhang, “Time series forecasting using a hybrid ARIMA and neural network model,” *Neurocomputing*, vol. 50, pp. 159–175, 2003, doi: 10.1016/S0925-2312(01)00702-0.

[4] Selectra for expats, *What is an Average Energy Bill in France?* [Online]. Available: https://en.selectra.info/energy-france/guides/tips/bills/average (accessed: Nov. 28 2020).

[5] KDnuggets, *6 Industries Warming up to Predictive Analytics and Forecasting - KDnuggets.* [Online]. Available: https://www.kdnuggets.com/2019/05/6-industries-warming-up-predictive-analytics-forecasting.html (accessed: Nov. 28 2020).

[6] Y.-H. Hsiao, “Household Electricity Demand Forecast Based on Context Information and User Daily Schedule Analysis From Meter Data,” *IEEE Trans. Ind. Inf.*, vol. 11, no. 1, pp. 33–43, 2015, doi: 10.1109/TII.2014.2363584.

[7] BrainKart, *Forecasting - Definition | Statistics.* [Online]. Available: http://www.brainkart.com/article/Forecasting_39271/ (accessed: Nov. 28 2020).

[8] K Sreenivasulu and G Mokesh Rayalu, “Goodness of fit tests for parameters of forecasting models,” *Int. J. Stat. Appl. Math.*, vol. 3, no. 1, pp. 470–475, 2018. [Online]. Available: https://www.mathsjournal.com/archives/2018/vol3/issue1/PartF/3-3-25

[9] K. Gajowniczek and T. Ząbkowski, “Electricity forecasting on the individual household level enhanced based on activity patterns,” *PloS one*, vol. 12, no. 4, e0174098, 2017, doi: 10.1371/journal.pone.0174098.

[10] A. Almalaq and J. J. Zhang, “Evolutionary Deep Learning-Based Energy Consumption Prediction for Buildings,” *IEEE Access*, vol. 7, pp. 1520–1531, 2019, doi: 10.1109/ACCESS.2018.2887023.

[11] K. Li, C. Hu, G. Liu, and W. Xue, “Building's electricity consumption prediction using optimized artificial neural networks and principal component analysis,” *Energy and Buildings*, vol. 108, pp. 106–113, 2015, doi: 10.1016/j.enbuild.2015.09.002.

[12] Thomas Linton, “Forecasting hourly electricity consumption for sets of households using machine learning algorithms,” 2015.

[13] A. M. Alonso, F. J. Nogales, and C. Ruiz, “A Single Scalable LSTM Model for Short-Term Forecasting of Disaggregated Electricity Loads,” Oct. 2019. [Online]. Available: https://arxiv.org/pdf/1910.06640

[14] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, “Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network,” *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, 2019, doi: 10.1109/TSG.2017.2753802.

[15] S. Ai, A. Chakravorty, and C. Rong, “Evolutionary Ensemble LSTM based Household Peak Demand Prediction,” in *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Okinawa, Japan, Feb. 2019 - Feb. 2019, pp. 1–6.

[16] A. Marvuglia and A. Messineo, “Using Recurrent Artificial Neural Networks to Forecast Household Electricity Consumption,” *Energy Procedia*, vol. 14, pp. 45–55, 2012, doi: 10.1016/j.egypro.2011.12.895.

[17] a. parate, “Individual Household Electric Power Consumption,”

[18] *PrePly_DNN.pdf - Deep Neural Network XXXXX XXXX November 8 2020 XXXX Machine Learning and Artificial Intelligence Its Application in t Outline Deep | Course Hero.* [Online]. Available: https://www.coursehero.com/file/72689528/PrePly-DNNpdf/ (accessed: Nov. 28 2020).

[19] TIBCO Software, *What is Deep Learning?* [Online]. Available: https://www.tibco.com/reference-center/what-is-deep-learning (accessed: Nov. 28 2020).

[20] J. Moolayil, Moolayil, and S. John, *Learn Keras for Deep Neural Networks*, 1st ed. [Place of publication not identified]: Apress, 2019.

[21] *Deep Learning Frameworks | NVIDIA Developer.* [Online]. Available: https://developer.nvidia.com/deep-learning-frameworks (accessed: Nov. 28 2020).

[22] J. Brownlee, “Your First Deep Learning Project in Python with Keras Step-By-Step,” *Machine Learning Mastery*, 23 Jul., 2019. https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/ (accessed: Nov. 27 2020).

[23] A. R. S. Parmezan, V. M. Souza, and G. E. Batista, “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*, vol. 484, pp. 302–337, 2019, doi: 10.1016/j.ins.2019.01.076.

[24] *Understanding Neural Networks. From neuron to RNN, CNN, and Deep Learning | by vibhor nigam | Towards Data Science.* [Online]. Available: https://towardsdatascience.com/understanding-neural-networks-from-neuron-to-rnn-cnn-and-deep-learning-cd88e90e0a90 (accessed: Nov. 27 2020).

[25] V. Nigam, “Understanding Neural Networks. From neuron to RNN, CNN, and Deep Learning,” *Towards Data Science*, 11 Sep., 2018. https://towardsdatascience.com/understanding-neural-networks-from-neuron-to-rnn-cnn-and-deep-learning-cd88e90e0a90 (accessed: Nov. 28 2020).

[26] KDnuggets, *Recurrent Neural Networks Tutorial, Introduction - KDnuggets.* [Online]. Available: https://www.kdnuggets.com/2015/10/recurrent-neural-networks-tutorial.html (accessed: Nov. 28 2020).

[27] CAIS++, *Lesson 8: Recurrent Neural Networks.* [Online]. Available: http://caisplusplus.usc.edu/blog/curriculum/lesson8 (accessed: Nov. 28 2020).

[28] WildML, *Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs.* [Online]. Available: http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/ (accessed: Nov. 27 2020).

[29] S. Elsworth and S. Güttel, "Time Series Forecasting Using LSTM Networks: A Symbolic Approach," Mar. 2020. [Online]. Available: http://arxiv.org/pdf/2003.05672v1

[30] S. Elsworth and S. Güttel, "Time Series Forecasting Using LSTM Networks: A Symbolic Approach," Mar. 2020. [Online]. Available: http://arxiv.org/pdf/2003.05672v1

[31] F. A. Gers and E. Schmidhuber, "LSTM recurrent networks learn simple context-free and context-sensitive languages," *IEEE transactions on neural networks*, vol. 12, no. 6, pp. 1333–1340, 2001, doi: 10.1109/72.963769.

[32] S. Ali, H. Mansoor, N. Arshad, and I. Khan, "Short Term Load Forecasting using Smart Meter Data," in *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, Phoenix AZ USA, 2019, pp. 419–421.

[33] *UCI Machine Learning Repository: Individual household electric power consumption Data Set.* [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption (accessed: Nov. 26 2020).

[34] Wikipedia, *AC power.* [Online]. Available: https://en.wikipedia.org/w/index.php?title=AC_power&oldid=984930043 (accessed: Nov. 26 2020).

[35] F. Nelli, *Python Data Analytics: With Pandas, NumPy, and Matplotlib*, 2nd ed. Berkeley, CA: Apress, 2018.

[36] R. Python, "Python Statistics Fundamentals: How to Describe Your Data," *Real Python*, 16 Dec., 2019. https://realpython.com/python-statistics/ (accessed: Nov. 26 2020).

[37] *Consumption – Sensitivity to temperatures and end-uses : RTE Bilan électrique 2019.* [Online]. Available: https://bilan-electrique-2019.rte-france.com/sensitivity-to-temperatures-and-end-uses/?lang=en (accessed: Nov. 28 2020).

[38] GitHub, *Pushpak2227/Master-Thesis.* [Online]. Available: 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: Nov. 26 2020).

[39] J. Qiu, B. Wang, and C. Zhou, "Forecasting stock prices with long-short term memory neural network based on attention mechanism," *PloS one*, vol. 15, no. 1, e0227222, 2020, doi: 10.1371/journal.pone.0227222.

[40] K. Grace-martin, "Assessing the Fit of Regression Models," *The Analysis Factor*, 08 Dec., 2008. https://www.theanalysisfactor.com/assessing-the-fit-of-regression-models/ (accessed: Nov. 22 2020).

[41] J. Brownlee, "Multi-Step LSTM Time Series Forecasting Models for Power Usage," *Machine Learning Mastery*, 09 Oct., 2018. https://machinelearningmastery.com/how-to-develop-lstm-models-for-multi-step-time-series-forecasting-of-household-power-consumption/ (accessed: Dec. 7 2020).

[42] GitHub, *Pushpak2227/Master-Thesis*. [Online]. Available: https://github.com/Pushpak2227/Master-Thesis/blob/master/Prediction%20for%20a%20week%20by%20taking%20past%207%2C14%2C21%2C28%20days%20values.ipynb (accessed: Nov. 26 2020).