Ministry of Higher Education and
Scientific Research
University of Diyala
Department of Computer Science

Analysis of Electrical Energy Consumption Data Set Based on Predication Models

A thesis

Submitted to the Department of Computer Science\ College of Sciences\ University of Diyala in a Partial Fulfillment of the Requirements for the Degree of Master in Computer Science

By

Zahraa Jabbar Zghair

Supervised By

Prof. Dr. Dhahir Abdulhade Abdulah

2020 A.C 1441 A.H

بسم الله الرحمن الرحيم

والذي هُوَ يُطْعِمُنِي وَيَسْقِينِ ﴿٧٩﴾

وَإِذَا مَرِضْتُ فَهُوَ يَشْفِينِ ﴿٨٠﴾

صدق الله العظيم

سورة الشعراء الاية (79_ 80)

Acknowledgment

First of all, praise is to GOD, the lord of the whole creation, on all the blessing was the help in achieving this research to its end.

I wish to express my thanks to my college (college of science), my supervisor, prof. Dr. Dhahir Abdulhade Abdulah for supervising this research and for the generosity, patience and continuous guidance throughout the work. It has been my good fortune to have the advice and guidance from him. My thanks to the academic and administrative staff at the Department of the computer sciences.

Zahraa Jabbar

Dedication

To ...

My father and my brother
Mohammed, may God have mercy
on them
My dear Mother
My husband Khalid for his
unlimited love, support, endurance
and encouragement
My candle, my children
My sister and my brothers
My friends
I produce this work with all my
love….

Zahraa Jabbar

(Linguistic Certification)

I certify that this research entitled "***Analysis of Electrical Energy Consumption Data Set Based on Predication Models***" was prepared by ***Zahraa Jabbar Zghair*** and was reviewed linguistically. Its language was amended meet the style of English language.

Signature:

Name: **Prof. Dr. Luma Ibrahim Al-Barazenji**

Date: 23 / 11 / 2020

(Supervisor's Certification)

We certify that this research entitled "***Analysis of Electrical Energy Consumption Data Set Based on Predication Models***" was prepared by ***Zahraa Jabbar Zghair*** Under our supervisions at the University of Diyala Faculty of Science Department of Computer Science, as a partial fulfillment of the requirement needed to award the degree of ***Master of Science in Computer Science.***

(Supervisor)

Signature:

Name: **Prof. Dr. Dhahir Abdulhadi Abdulah**

Date: 23 / ١١ /2020

Approved by University of Diyala Faculty of Science Department of Computer Science.

Signature:

Name: **Assist. Prof. Dr. Taha Mohammad Hassan**

Date: 27 / 11 /2020

(Head of Computer Science Department)

Abstract

The rising population growth and the number of electrical appliances used day by day, leads to an increase in the consumption of electrical energy, hence the demand for electricity, leading to strain on electricity suppliers. Because there are many factors that affect electricity consumption, the use of the smart meter technology enables us to obtain massive amounts of data around the clock, this facilitates predicting power consumption and energy management regulation.

In this study, a model was proposed to predict the energy consumption of a single household in the short term (one day, one week) and medium term (one month). The proposed model consists of four stages: data collection, pre-processing stage, the prediction stage and the performance evaluation stage. Data set was collected through a single house smart meter for model validation and results analysis. Then, a deep learning machine Long Short-Term Memory LSTM, and well-known machine learning algorithms Support Vector Regression SVR, K-nearest neighbor KNN and Naive Bayes applied it to pre-processed data to predict energy consumption for one day, one week and one month. They are compared using statistical measures: Mean absolute error (MAE), mean absolute percentage error (MAPE) and root mean square error (RMSE) for performance measurement of these machine learning algorithms.

These statistical measurement values indicate that the performance of proposed model LSTM is better than K-NN, SVR and Naïve Bayes for predicting energy for one day, one week and one month on the data given. The results of the model LSTM are MAE 0.183, MAPE 18.324 and RMSE 0.244 for one day power consumption prediction, MAE 0.145, MAPE 15.182 and RMSE 0.179 for one-week power consumption prediction. MAE 0.145, MAPE 14.018 and RMSE 0.166 for one-month power consumption prediction. It is clear that LSTM model is capable of predicting the consumption of electricity in the short term (one day, one week) and medium term (one month) with high accuracy.

List of Contents

Chapter Three: The Proposed Model

Chapter Four: Experimental Results and Evaluation

List of Abbreviations

Abbreviation	***Description***
Adam	*Adaptive Moment Optimizer*
AMI	*Advanced Metering Infrastructure*
ANN	*Artificial Neural Network*
DNN	*Deep Neural Network*
GP	*Genetic Programming*
KNN	*K-nearest-neighbor*
KWh	*Kilowatt-hours*
LSTM	*long Short-Term Memory*
MAE	*Mean Absolute Error*
MAPE	*Mean Absolute Percentage Error*
MR	*Multiple Regression*
PDF	*Probability Density Function*
QoS	*Quality of service*
RBF	*Radial Basis Function*
RMSE	*Root Mean Square Error*
RNN	*Recurring Neural Network*
SG	*Smart grid*
SMEs	*Small and medium size enterprises*
SVM	*Support Vector Machine*
SVR	*Support Vector Regression*

List of Figures

List of Tables

List of Algorithms

Chapter One

Introduction

Chapter one

Introduction

1.1 Overview

Consuming power is an everyday practice for people around the world that is done with no real care. Because of the rapid economic development and increasing population growth, the previous decades witnessed a steady high demand for energy usage and thus consumption. As a result, demand on power exceeded the generation capacity leading to difficulties to meet that high demand in some part of the world. The management of energy consumption problem is too big to deal with the losses caused by the growing consumption patterns [1].

The rapid increase in energy consumption requires an accurate expectation of the distribution of electricity consumption [2]. In order to accurately predict the use of electricity, it is necessary to track electricity consumption. Therefore, Advanced Metering Infra-structure AMI was introduced. AMI leads to a large amount of energy consumption data. AMI data is used to predict energy consumption. The prediction helps make decisions about energy distribution from the national grid. Accurate forecasting of electricity consumption can prevent unplanned power outages [3].

The smart grid (SG) is considered one of the most important applications of the Internet of Things. It is an integrated data communication network that is used to collect and analyze data via transmission lines and distribution substations as well as the final consumer throughout the electricity network, and an expectation of consumption can be obtained through this data provided to energy suppliers for strategies Efficient energy management [4].

The very crucial part of the smart grid to register power consumption all over the power grid at the end user side, is the smart meter that collect data on an hourly basis or less and feedback such data to suppliers. Smart meters allow contact between the meter and the central network possible in two ways. The collection of detailed data in an interval of 15 mints or less will assist power suppliers as well as consumers to have a comprehended view of the consumption patterns via data analytics, which has become an important part of the industry research and development field [5]. The data (meter data analytics) sent by smart meters are analyzed for the purposes of:

- Utilization of usage patterns to help in decision making pertaining to purchases.
- Making power consumption predictions using via previous consumption patterns.
- Maintaining efficient power supplies in cooperation with consumers.
- Finding out illegitimate grid connections.
- Compare and correct the performance of meter service providers, to reduce unpaid bills and better maintenance decision to help keeping the grid on [1].

Electricity consumption is a time-dependent attribute. Therefore, there are approaches that use time series to build the model to predict electricity consumption. Availability of past information leads to solutions based on time series analysis since it reflects the time-dependent variations [6].

The forecasts for electricity consumption have been identified as short term (hourly to one week), mid-term (one week to one year), and long term (more than one year) forecasts [7].

Time-series analysis techniques are addressed using conventional approaches and artificial intelligent-based approaches (Artificial Neural Network (ANN), Deep Neural Network (DNN), Multiple Regression (MR), Support Vector Machine (SVM), Genetic Programming (GP)). There are many challenges for mid-term and long-term electricity consumption forecasting [3].

This thesis presents treatment of increased electricity consumption through proposed energy consumption prediction model that use four approaches, along Short-Term Memory (LSTM), a Support Vector Regression (SVR), a K-nearest-neighbor (KNN) and a Naive Bays through forecast electricity consumption for short-term (one day, one week), mid-term (one month). The reason for using these algorithms is that they are a great fit for our problem, since electricity consumption is constantly variable.

1.2 Related Works

There is a group of studies and researches in this field that dealt with methods of predicting power consumption through the use of smart meter data and the most important are:

- **Zheng et al,2017 [8].** The use of a long-term-short-term memory (LSTM)based repetitive neural network (RNN) has been proposed to address the short-term electrical load prediction problem, using a long-term electricity consumption data set. And compare it in the following methods: SARIMA (Autoregressive Integrated Moving Average) which is a seasonal moving average model with integrated automatic regression, NARX (Nonlinear Autoregressive Network with Exogenous inputs) which is a nonlinear neural network model with external input, SVR (Support Vector Regression) which is a very popular model in financial time series prediction and NNETAR (Neural NET work Auto Regression) which is an automatic neural network model to predict single-variable time series with single-layer hidden and lagging inputs. Two evaluation criteria were used as a measure of performance: root mean square error (RMSE) and mean relative absolute error (MAPE) between real values and prediction results. Results showed that LSTM outperforms all other methods with the best expected time series. the results for LSTM are RMSE =0.0702 and MAPE =0.0535.

- **Quek et al, 2017 [9].** Proposed a short- term forecasting method that applies Naïve Bayes Classification (NBC) machine learning technique on easily available input parameters are considered as the continuous-valued data such as instantaneous power, outdoor temperature, panel temperature, on-site irradiance and time of the day to predict the overall energy generated by photovoltaic cells which are installed in distributed region in the next 15-minute period. Based on rule-based inferences, continuous-valued data is converted into categorical-valued data. Categorical valued data is represented as class labels like ‘very high’, ‘high’, ‘medium’, ‘low’, ‘very low’. Historical test data of an existing photovoltaic system located in Singapore is used to evaluate the accuracy of the NBC forecasting method and the comparison demonstrates that the proposed method is able to achieve a forecasting accuracy of over 68 percent.

- **Fayaz et al,2018 [10].** Proposed a methodology for predicting energy consumption in apartment buildings. The proposed method consists of four different layers, namely data acquisition, pretreatment, forecasting, and performance assessment. For experimental analysis, they collected real data from four multi-storied apartment buildings. This data is collected as input to the acquisition layer. In the pre-processing layer, several data cleaning schemes have been published to remove anomalies from the data. In the prediction layer, a Deep extreme Learning Machine (DELM) is used to predict energy consumption. In addition, the use of the adaptive neuro-fuzzy inference system (ANFIS) and the Artificial Neural Network (ANN). A different number of hidden layers, different hidden neurons, and different types of activation functions have been used in DELM to achieve the optimal DELM structure for predicting energy consumption. In the performance appraisal layer for the comparative analysis of three prediction algorithms, mean absolute error (MAE), root mean square error (RMSE) and mean absolute percentage error (MAPE) were used. The results indicate that DELM performed significantly better than ANN and ANFIS for predicting energy for one week and one month on the data given. The results of the algorithm DELM are MAE 2.0008, MAPE 5.7077and RMSE 2.2451 for one-week prediction. MAE 2.3347, MAPE 6.5464 and RMSE 2.6864 for one-month prediction.

- **Gokgoz F. et al,2018[11].** In this study, models were presented based on deep neural networks, especially long short-term memory algorithms LSTM to predict renewable energy loads with a short-term forecasting horizon, by using data models from the mechanism to support renewable energy resources in Turkey. With an accuracy of one hour between January 2016 and December 2017. Creating 432 different models by changing the cell number of layers and leakage. Instead of SGD Stochastic gradient descent (random gradient) the “adaptive torque estimation” algorithm used for training as a gradient-based optimizer. It performed better than SGD in terms of speed in convergence and lower error rates. Absolute mean error (MAE) and square mean error (MSE) were used to compare model performance. Of the 432 models, five results for MAE were 0.66, 0.74, 0.85, and 1.09.

- **Zhang et al,2018 [12].**In this study, the "support vector regression" (SVR) modeling approach was used to predict the consumption of the individual electric family applied to daily and hourly data for the use of electricity for fifteen households from 2014 to 2016, and by using different methods to divide the dataset into a subset of training and testing for families that are similar in electricity consumption over time, as the successive division on the basis of time works better than randomly sampled data. As for families that lack regularity in the hourly electricity use, then randomly sampling data and using 20% of them as a sub-data set test outperforms the existing approach on the time, since the accuracy of daily data achieved the results of forecasting the best hourly data for all households. Using mean absolute percentage error (MAPE) for one of the fifteen households, the daily forecast is 12.78 and the hourly forecast is 23.31, and it drops to 22.01 (per hour) if only weekdays are calculated for the same the family.
- **George et al, 2018 [13].** Proposed an analytical model describing energy consumption by using energy profiles, which gives energy to the consumer over a period of time, to conduct quantitative analysis using smart meters section. This consumer of the same type assembly and the number of devices together section. This use K-Means algorithm and k-nearest neighbor classification, the value of the account consumption of the most efficient and average consumption within each cluster on a monthly basis and used these accounts to compare individual user consumption, the use of the data set containing the monthly electricity consumption for each apartment for one year.

- **Kim and Cho,2019 [14].** Proposed model that combine the convolutional neural network and long short-term memory (CNN-LSTM). The proposed model extracts spatial and temporal features to actively predict energy consumption for the individual household electric power consumption dataset. Long-term, mid-term, short-term forecasting and real-time forecasting were considered by aggregating energy consumption in units of minute, hourly, daily, and weekly. Linear regression and LSTM models were used to compare experimental results. To evaluate performance, mean squared error (MSE), root mean square error (RMSE), mean absolute error (MAE) and mean absolute error ratio (MAPE) were used. The proposed method achieves higher performance than linear regression and LSTM, with RMSE = 0.6114,0.5957, 0.3221, 0.3085 respectively for energy consumption per minute, hourly, daily and weekly.
- **Adewuyi et al,2020 [15].**They applied three models of deep education which are (MLP), (CNN) and (LSTM) to predict electricity demand in the short term by using consumption data at the university in addition to data on the effect of weather on loads in the tropics, and compared them with each other using RMSE,MSE and MAE to measure the accuracy of the prediction As it was measured during the stages of testing, evaluation and training on different epochs 100, 80, 60, and 40, the results showed that the LSTM model outperformed the rest of the models. It was among its results in the test scale of the epoch 80, RMSE=2.46, MSE=0.45 and MAE=2.44.
- **Solyali D., 2020[16].** In this study, the techniques of artificial neural network (ANN), the Adaptive Neuroscience System (ANFIS), multiple linear regression (MLR), and the support vector machine (SVM) were used to predict the electrical load in Cyprus. Historical data were used to show the use of electricity for the period 2016-2017

with long and short- term analysis in Cyprus, and the parameters were temperature, humidity, population, electricity price per kilowatt hour, gross national income per capita, solar radiation. The results indicated that the support vector regression (SVR) is relatively superior to other models, as it showed lower prediction errors (4.34%, 4.49%) and root mean square error (RMSE) (25.43, 26.44) for long-term prediction. In the short-term term, artificial neural network (ANN) techniques showed better results than other techniques with lower prediction errors (0.97% and 1.67%) and root mean square error (RMSE) (7.67, 14.91).

Table (1.1): Related Works Summarizations

No.	year	Author	technique	accuracy

1	2017	Zheng et al. [8]	LSTM	RMSE=0.0702
				MAPE =0.0535

2	2017	Quek et al. [9]	NBC	68%

3	2018	Fayaz et al. [10]	DELM	RMSE=2.2451
				MAE =2.0008
				MAPE= 5.7077

4	2018	Gokgoz F. et al. [11]	LSTM	MAE=0.66
5	2018	Zhang et al. [12]	SVR	MAPE=12.78
6	2018	George et al. [13]	K-Means KNN	———
7	2019	Kim and Cho [14]	CNN-LSTM	RMSE=0.3221

8	2020	Adewuyi et al. [15]	LSTM	RMSE=2.46
				MSE=0.45
				MAE=2.44

9	2020	Solyali D. [16]	SVR
			ANN
				prediction errors =4.34%,
				RMSE=25.43
				prediction errors =0.97%
				RMSE=7.67

1.3 Problem Statement

The rapid increase in human population and development in technology have sharply raised power consumption in today's world. Since electricity is consumed simultaneously as it is generated at the power plant, it is important to accurately predict the energy consumption in advance for stable power supply. Peak demand is a problem that the power industry has ever faced as it requires more cost-effective and efficient procedures rather than adding more generators. And since accurate electricity consumption forecasts are of utmost importance in energy planning, they provide strong support for effective energy demand management. This work demonstrates the possibility of using the best model to obtain the best predictive energy consumption. This will reduce the gap between consumers and energy facilities so that they can Communicate more efficiently.

1.4 Aim of The Thesis

This thesis aimed to predict energy consumption using smart meter data through a case study on a single house and choosing a good forecast model for predicting energy consumption.

1.5 Objective

To contribute to reducing energy consumption, changing people's opinion a little smarter during the day in order for a better distribution of energy consumption. and prevent unplanned power outages.

1.6 Thesis Organization

Beside this chapter, the remaining parts of this thesis includes the following chapters:

Chapter Two: Theoretical Background

It presents an extensive overview of evolution of meters from past, traditional meters. It discusses the motions of Smart Grid, Smart Meter, power consumption, prediction of consumption. Also, it illustrates the basic principles and the scientific theories of the analysis data.

Chapter Three: The Proposed System

This chapter introduces the steps of the proposed prediction system, with its design and implementation.

Chapter Four: Experimental Results and Evaluation

This chapter presents the experiments and the results which are obtained from the system running and evaluates these results.

Chapter Five: Conclusions and Suggestions for Future Work

This chapter presents the conclusions of this work. Furthermore, it provides suggestions for future work.

Chapter Two

Theoretical Background

Chapter Two

Theoretical Background

2.1 Introduction

This chapter sheds light on the aspects of the theoretical aspects of analyzing smart meter data and predicted the electrical energy consumption. It includes the development of electricity meters, the conventional electricity meters, smart grid, the consumption of power, prediction of consumption, data normalization, long and short-term memory (LSTM) algorithm, k-nearest-neighbor algorithm (KNN), support vector regression (SVR), Naive Bayes algorithm and accuracy metrics.

2.2 The Development of Previous Electricity Meters

Electricity was available in the early years only to a particular section of the developed nations. However, this has changed with the development of technology as it has also taken the needs of average people in different parts of the world. When the electric meters are concerned, one can see that it includes different researchers from the past. For example, until the early 1870s, the general use of electricity was restricted to arc lamps and telegraph. Nonetheless, Thomas Elva Edison's invention of the electric lamp had a great impact on the power-energy market-making widely available to the public in 1879. In the year 1888, Oliver B. Shallenberger invented an AC ampere-hour meter leading to a remarkable evolution in the metering technology which in turn brought lights to households of numerous common people [17].

2.3 Traditional Electrical Meters

The electrical instruments, which are capable of measuring and showing energy interpretations, are called electricity meters. Those Standard meters have been employed since the late 19th century. Aluminum disks are used in most of the conventional electric meters to determine the usage of power [18]. Compared to nowadays, electricity meters are operated digitally which also have their limitations. Figure (2.1) shows a basic 1 phase 2 wire electricity meter.

Figure (2.1): Traditional meter [19]

The limitation of traditional electrical meters [18] can be viewed as the following:

•The nature of the meters is inefficient because users have to expect the monthly bill for electricity.

•The measurement mechanism is assisted by a particular mechanical system and is thus named electromechanical meters.

•A large number of employed inspectors are needed to perform meter readings.

•The handling of the payments is costly and time-consuming.

•New forms of tariffs cannot be imposed on an hourly basis with the accompanying meters for customer motivation.

•The Complicated development of meter program systems and the complicity of supporting network infrastructure.

In addition, other shortcomings of electrical meters can be seen. For example, the instalment of traditional meters leads to unsatisfactory user experience. Moreover, meters are of varying types. Thus, although timely electricity meter development helps the consumer gain knowledge regarding electricity consumption, consumption statistics could not be altered [18].

2.4 Smart Grid (SG)

One of the recent developments of electrical grids is smart grids. There are a number of factors that affect the efficiency of recent electrical grids. For example, they are becoming poor regarding the variability in the electrical load of domestic appliances. Furthermore, population growth is another indication that shows that electric grids are becoming less unreliable. The efficiency of electrical grids is determined by population growth. Enhancing the grid performance can be maintained by remotely monitoring and increasing reliability, measuring utilization in a communication assisted by providing data (real-time) to customers, suppliers and the other way round is named the Smart Grid [20]. Smart grids utilize automated sensors. Such devices are responsible for transmitting the assessed data back to utilities and have the potential to transfer power faults and prevent power line heating. This uses the self-healing feature. Practically, there is a direct link between smart meters and smarts grids. By 2030, its installations are expected to reduce carbon emissions by 5 per cent annually and it can have a positive influence on climate changes [21]. As a result, Smart Grids are recommended for many countries for sustainable development and the construction of modern grid networks.

Figure (2.2) shows Smart grid (SG) architecture presenting power systems, power flow and information flow. The SG is comprised of four main subsystems (power generation, transmission, distribution and utilization) and three types of networks (a wide area network (WAN), a neighborhood area network (NAN) and a home area network (HAN). The power flows through the subsystems while the information flows through the networks [22].

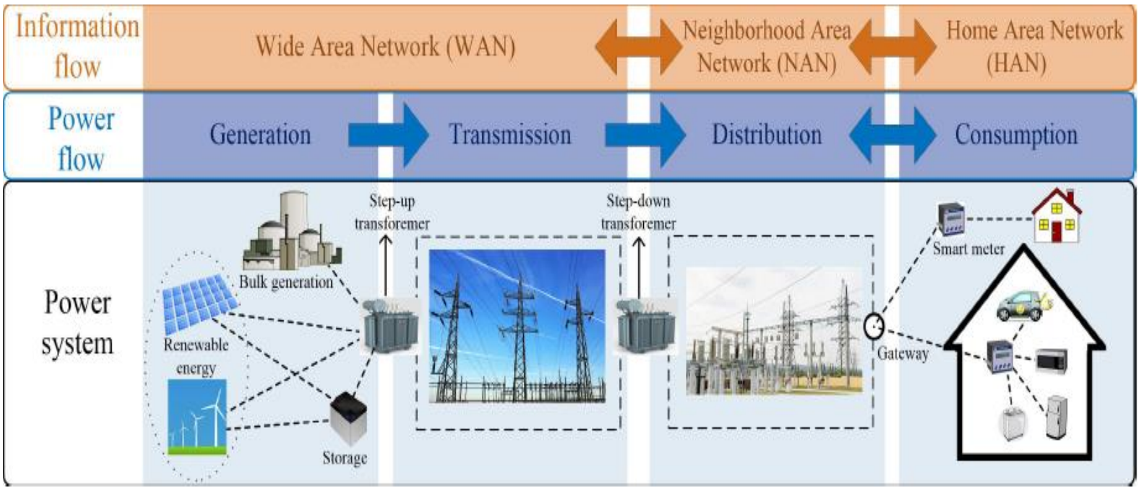


Figure (2.2): Smart Grid (SG) Architecture [22]

2.5 Smart Meter

A great feature of Smart Meters is that they are taking the consumption of energy into consideration; therefore, are used for KWh (Kilowatt-hours) calculation of electrical energy [23]. Thus, it can be said that this is a tool that gives customers who wish to save money on their power bill a direct advantage.

Figure (2.3): Smart Meter [24]

The smart meter should offer reliable meter reading with the inclusion of firm advantages. They also record the use based on intervals of hours or fewer than an hour. A smart meter has non-volatile data storage capabilities, remote connection or disconnection capabilities, finding tamper, and communicating in a two-way technology called advanced integrated metering (AMI). Furthermore, they send the gathered data directly to the central meter. The main meter controls the smart meter 's functionality. The use of smart metering from an operational perspective allows control and manage the electricity grid in a better way [25]. Some of the advantages that smart meters offer are:

•Low running costs.

•Saving time for customers and utility suppliers to send the meter reading back to their electricity providers.

•Online payment of electricity bills is allowed.

•The use can be greatly reduced with an intimation policy during the high peaks.

•Has the automated termination feature when the devices are not in use [26].

While the term smart meter only began to be used after the SG initiatives, it can be seen that meter features and functionality evolved from the previous manually read meters to AMI meters with dashboard interfaces and two-way communication capabilities [25].

Smart meter senses all the in-resident consumption produced. Meter measurements provide the energy providers with a better perception such that the habits of the inhabitants' total consumption of electricity can be changed [27].

2.6 The Consumption of Power

The overall volume of electricity used in an individual home is referred to as the energy consumption. Power use is an essential part of supplying electricity. Thus, citizens should be conscious of the electricity that need to be saved for potential use in the long run. The energy levels were changing gradually with the regular use of electricity. Such change in usage habits can be caused by environmental conditions or occupants' excessive use of electricity such as a rise in household equipment use and reckless behavior of using, for example, not turning off the lights or television while not viewing. Such variables can have higher effects on end-users. Considering that the power supplied by energy firms is enormous, most people ignore resources and savings. In the mentality of utilities, the value of consumption is increasing. Energy services should play a significant part in promoting the smart meter system and should engage people in rising energy impacts by increasing awareness of the effects of their actual usage rates [28].

Figure (2.4): Smart Meter Gaging Electrical Machines in a house [29]

Figure (2.4) illustrates the day-to-day operations of household appliances calculated in a home by a smart meter. Inside the building, the smart meter is installed where usage data is analyzed and utilized to build a predictive energy demand management system and reduce consumer electricity bills. This measuring facility transforms a simple home into a smart home [30].

AMI is an improved and modified version of automatic meter reading. Information from various types of meters is automatically collected in an automatic meter reading and transmitted to a central database for future review and billing purposes through a one-way communication network. AMI was introduced because the fully automated meter reading could not provide bidirectional communications [31].

AMI components are made of a centralized system, two-way networks of communication, data concentrations, and smart meters. Small and medium size enterprises (SMEs) are installed at customer premises or other smart grid positions to calculate usage data and send it to the centralized computer through billing communications systems, reminding customers about their usage, etc. smart meters may offer power usage overviews indirect load management, and schedule times for switching on and off machines to change the load in SG. Direct load management can also add distributed energy services to SG to provide higher load while the power grid produces additional energy [32].

2.7 Prediction of Consumption

Electrical energy has an important role as it is used regularly throughout the world and because it cannot be stored and used later, but it must be generated and transferred based on demand, so forecasting the demand for electricity is the way in which the supplying companies expect the energy that the consumer needs and providing the required load demand on the short, medium and long term, where forecasting has an important role in distributing loads and planning for the construction of future generation facilities. The consumption prediction according to the forecast period can be divided into:

Short term: usually from one hour to one week.

Medium term: expectations are usually from several weeks to several months.

Long term: expectations are from one to several years [7,33].

In this thesis, the forecasting consumption will be discussed in the short term (one day, one week) and the medium term (one month).

2.8 Data Normalization

One of the techniques of data pre-processing is normalization, where attributes of a dataset are converted by measuring their value to fall within a specified small range such as 0.0 to 1.0. Minimum and maximum, z-score and decimal normalization are normalization techniques [34].

- **Min-max Normalization**

Min-max normalization leads to a linear transformation of the main data, and maintains the relationships between the original data values. Assume that min_A is the minimum value for attribute A, and max_A is the maximum value for attribute A. Min-max normalization maps a value v_i of A to v'_i in the range $[new\_max_A, new\_min_A]$ Calculated as equation (2.1)[34]:

$$v'_i = \frac{v_i - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A \qquad (2.1)$$

It will encounter an "out of bounds" error if the normalized future input state is outside the original data range of A [34].

2.9 Recurrent Neural Network

Recurrent Neural Network, or commonly identified as RNN are intended to collect data related to time series information. They receive varying number and send the output of adjustable magnitude, which works very well for data from time series. It can be challenging to comprehend RNN and much representation may lead to misunderstanding it. RNN operates as given in equation (2.2) [35]:

$$S_t = F_w(S_{t-1}, X_t) \tag{2.2}$$

The new state of the recurrent neural network at time t is a function of its old state at time t-1 and the input at time t. St = current state (at time t), S_{t-1} = previous state, and X_t= input at time t. This function is the basic idea behind RNN. The simplest implementation of RNN is presented in figure (2.5).

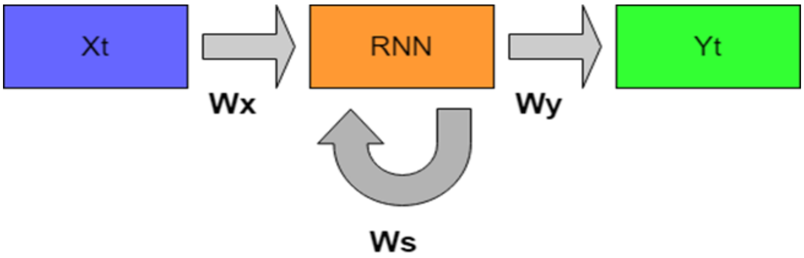


Figure (2.5): Simple RNN [36]

The recursive function is a tanh function which maps all values between -1 and 1, as described in equation (2.3) [35]. Where the input state is multiplied with the weight of input Wx and the previous state with Ws and pass it through a tanh activation which results in the new state as given in equation (2.4) [35]. In order to get the output vector, the new state is multiplied with that is St with W_y as given in equation (2.5) [35].

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.3)$$

$$S_t = tan\, h(W_s S_{t-1} + W_x X_t) \quad (2.4)$$

$$Y_t = W_y S_t \quad (2.5)$$

If RNN is unrolled following figure (2.6) it can be seen that there is the previous state S0, input at time step 1 is X1, these go into RNN and RNN calculate the next state based on its recursive formula tanh($W_s S_0$+$W_x X_1$) and gives us the state 1(S_1) and to get the output S_1 is multiplied with W_y.

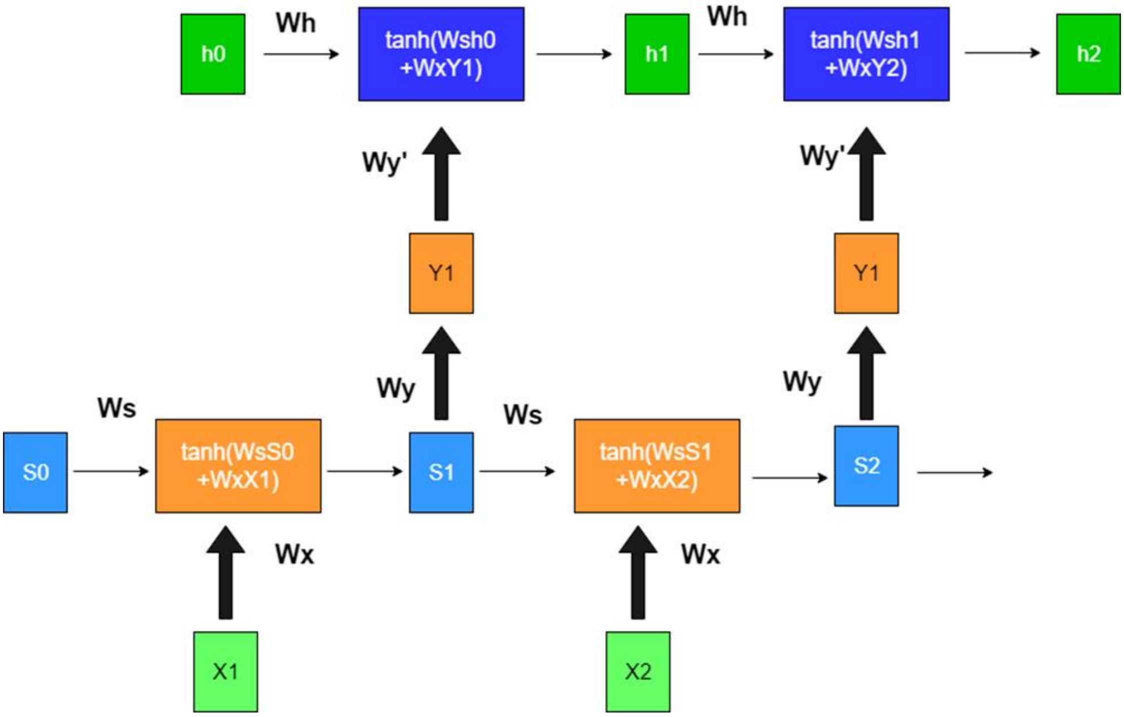


Figure (2.6): Simple RNN Unrolled [36]

The new state S_1 and input X_2 is the input for the new time step. get $S_2$and its output can be obtained by multiplying it with W_y. Nevertheless, the problem is that the same set of weight is used throughout the model. In the case of multilayer RNN, we serve outputs as the input of our next layer. HereY_1 and Y_2 act as input in the next layer. Furthermore, deeper networks usually offer better accuracy but we don't go any further into RNN. In general, people use deep modules with 2 to 3 layers. Over time, RNN learns to use backpropagation. Thus, the loss is calculated using the output and go back to the state by multiplying the gradient to change the weights. However, the weight update is nearly nil which is negligible. In other words, the suggested model would not acquire new data. This called the vanishing gradient problem[35]. More interactions can be added to RNN to solve and improve it, which is the idea behind LSTM.

2.9.1 Long Short-Term Memory (LSTM) Algorithm

LSTM solves the problem of vanishing gradients. LSTM cells can solve this issue by integrating memory cells into the RNN concealed layer. LSTM has acquired prominence in forecasting time series data for all the above reason.

The calculation formulas[37] related to the LSTM structure are:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{2.6}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{2.7}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{2.8}$$

$$\mathrm{C}'_t = tanh(W_{\mathrm{g}}[h_{t-1}, x_t] + b_{\mathrm{C}'}) \tag{2.9}$$

$$C_t = C_{t-1} * f_t + \mathrm{C}'_t * i_t \tag{2.10}$$

$$h_t = \tanh(C_t) * o_t \tag{2.11}$$

In the above-mentioned equations [37], the W_f, W_i, W_g, W_o are the corresponding weight matrix connecting the input signal $[h_{t-1}, x_t]$. While the b_i, b_f, b_o and $b_{C'}$ are the bias terms for each gate. The σ and tanh respectively represent Sigmoid and Hyperbolic tangent activation function.

The variables signify the following meanings f_t=Forget gate, i_t=Input gate, o_t=Output gate, C'_t=Intermediate cell state, C_t=Cell state andh_t=New State. LSTM 's advantage is that it has three gateways. These gates and cell states are supplementary interactions. Where there is the gate of forgetting which includes a logistic function sigmoid σ as described in equation (2.12) [37], and this gate takes the old state and input and increases it by the corresponding weight. Then it is passed through a sigmoid activation. Then the gate calculates a temporary output between 0 and 1, according to equation (2.6) [37].

$$sigmoid(x) = \frac{1}{1+e^{-x}} \qquad (2.12)$$

There are gate input and gate output, and they function by the same mechanism. The significant point is that every gate has different weight sets.

Again, C'_t is an intermediate cell state that is used to measure the cell status. Then the new condition is obtained by multiplying the cell state's tanh activation with the output gate [37]. This method can be represented using Figure (2.7).

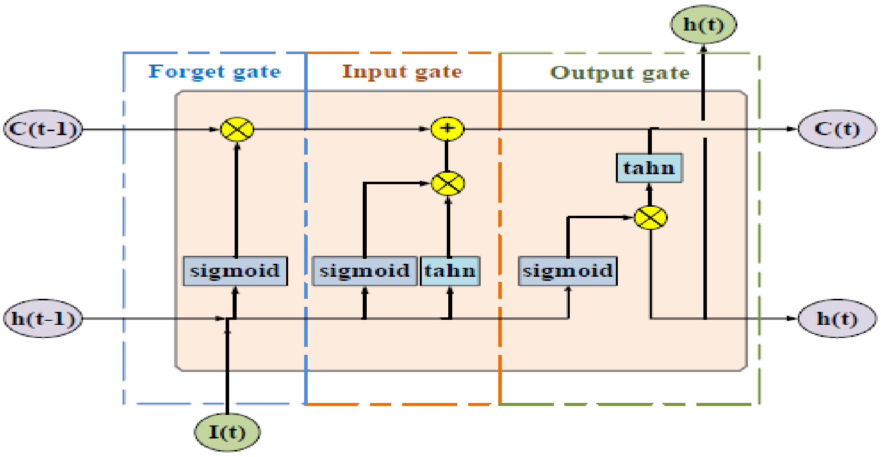


Figure (2.7): LSTM Mechanism [37]

2.9.2 Loss Function

It is also identified as the cost function, tests the consistency between network production forecasts through forwarding propagation and given ground truth markers. This function is usually used for multiclass grouping is cross-entropy, while usually mean squared error as given in equation (2.13) [38] is added to continuous value regression.

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(Actual_t - Predicted_t)^2 \qquad (2.13)$$

where n refers to the number of samples in the input set, MSE was used as the Loss function during the training phase to minimize the errors[38].

2.9.3 Optimizer

Optimization algorithms are responsible for minimizing or maximizing the error/loss function stated in the section above and update the system weights. There are different optimization algorithms, including RMSprop, Adam, Adamax, etc. Through experiments, it was found that the Adaptive Moment Optimizer (Adam) is best suited to solve such problems. Where in addition to using adaptive learning rates the parameter is based on the average of the first moment (mean) as in RMSProp, Adam uses the mean of the second moments of the gradient (non-composite variance).

Because Adam achieves good results in a short period of time, it is considered one of the most effective algorithms in machine learning. The translator requires the neural network training group to define both the optimization algorithm and the loss function [39,40].

2.9.4 Overfitting

This indicates to a condition in which a model acquires descriptive statistics patterns for the exercise set, i.e. eventually learning the unrelated noise instead of knowing the signal and thus has a lower performance on a corresponding new dataset. It is one of the main glitches in the area of machine learning, as an overfitted model is not generalizable to data that have not been seen before. In this context, this test is a key role in proper testing the results of machine learning models [41].

2.9.5 Dropout

It is a mechanism used in deep neural networks to enhance the training area 's performance and get rid of the negative Overfitting phenomenon, and it sometimes results from the wrong values. The word "dropout" refers to the units (hidden and visible) being lowered in a neural network. By dropping a unit out, it is removed from the network temporarily with all its input and output connections. These cells are randomly selected and do not use this technique in the stage of prediction and conclusion of no longer needed [42].

2.10 K-Nearest Neighbor Algorithm (KNN)

K-nearest neighbors' algorithm (k-NN) is a method that is utilized in machine learning for statistical organization and regression analysis. It is a procedure that simply saves the available cases and then models the new input data or case based on measuring its resemblance with the other cases. The K in KNN represents the value of nearest neighbors that will be used to foresee or label the new one. It is a hyperparameter that should be chosen in order to find the best conceivable t-value for the data. A hyperparameter is a parameter that cannot be derived directly via the regular process of training. In fact, K determines the decision boundary [43].

Regarding using KNN algorithm in regression analysis, the average value of K nearest neighbors is calculated in order to label the new data. KNN algorithms are determined by points such as the value of neighbors, and the implemented type of distance. K is one of the most crucial aspects that can have a huge impact on the quality of predictions. That is a minor value of K can lead to great alteration in foreseen outcomes in any given problem. On the contrast, using a big value of K will result in a huge bias in the model. Therefore, an appropriate value should be picked for K and it should be neither too outsized nor tiny.

Figure (2.8) allows us to understand how the K parameter picks the training set and forecasts unlabeled value. The objective here is to find the suitable value of k by using cross-validation in order to create a prototypical that is optimal.

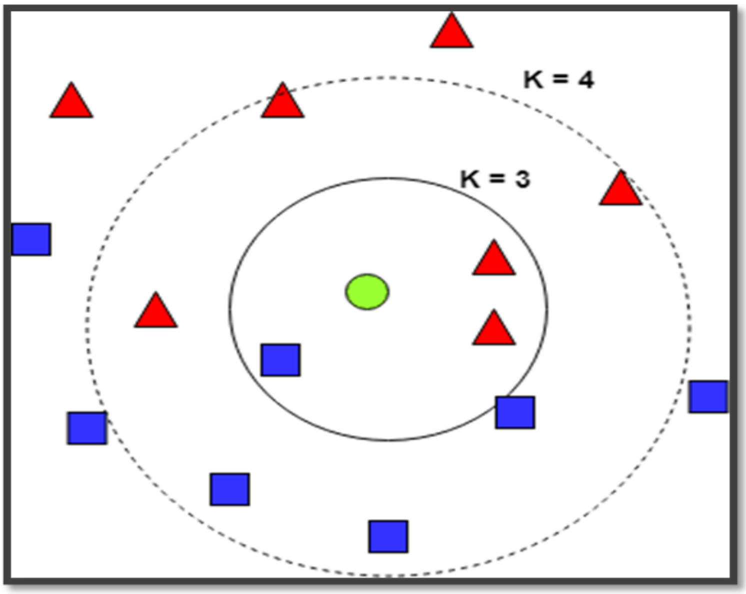


Figure (2.8): Working Principle of K Parameter [36]

As is illustrated in the previous paragraphs, KNN regression work consists simply of holding a set of training instances. For example, the i-th training instance is formed of a vector that has n features: (f_1^i, f_2^i, . . ., f_n^i), which, in its turn, describes the instance and an associated target vector that has m features: (t_1^i, t_2^i, . . .,t_m^i). So, a new instance is introduced, this instance's features are known (q_1, q_2, . . ., q_n), while its target is unknown. Based on the vectors of features and the measurement of similarity or distance, closest K value is found from the exiting training instances by using the features of the newly introduced instance. As an example, presuming that the Euclidean distance is the similarity metric, in this case, the distance between the new instance and the i-th training instance would be calculated by using the equation (2.14) [44]:

$$\text{Euclidean Distance } (f; q) = \sqrt{\sum_{x=1}^{n} (f_x^i - q_x)^2} \qquad (2.14)$$

The existing k training instances that are found to be the most similar to the new instance are regarded as their K closest instances or K nearest neighbors. KNN is carried out on the basis of learning by analogy. It is assumed that the targets of the newly-introduced instance are likely to be similar to its nearest neighbors. Meaning that the targets of the nearest neighbors are collected in order to predict the unknown target of the introduced instance. For instance, presuming that the targets or the k nearest neighbors of the new instance consists of the vectors: t_1, t_2, ..., t_k, the average of the vectors trying to predict the target of the new instance can be calculated by equation (2.15) [44]:

$$\text{Average(t)} = \sum_{i=1}^{k} \frac{t^{i}}{k} \qquad (2.15)$$

Briefly, the KNN algorithm consists of saving a set of training instances that are described by n features. Each training instance is represented by a point in an n-dimensional space. When a new instance is introduced, KNN locates the closest available k value in the n-dimensional space, hoping that their targets are similar to the unlabeled target [44].

2.11 Support Vector Regression (SVR)

Support vector regression (SVR) is a variant of Support vector machines (SVM). SVM are machine learning algorithm models that are used for problem regression and classification problems [45]. SVMs consist basically of a kernel algorithm and an optimizer. This works when Kernel divides ono-linear data into high-dimensional space and making data linearly separable. The learning takes place in the feature space, and the data points only appear inside dot products. Thus, the optimizer algorithm is applied to solve the optimization problem. Because SVM aims to reduce an upper limit of the generalization error consisting of the amount of the error rate and a degree of confidence, it demonstrates dominance relative to the widely used theory of empirical risk minimization (ERM), which minimizes only the error rate. Thus, SVM theory utilizes higher performance in generalization than other techniques in machine learning.

SVR is a nonlinear regression framework that looks at the poles of sets of data and makes a boundary judgment (or hyperplane) to address function fitting issues. A nonlinear analysis with an intense band of epsilon is seen in Figure (2.9). Often sets of data are linearly non-separable and need to be converted to an N-dimensional space and a splitting hyperplane (N-1)-dimensional needs to be found. The process, however, is computationally costly. An appropriate trick to the kernel could significantly reduce the computational expenses [45].

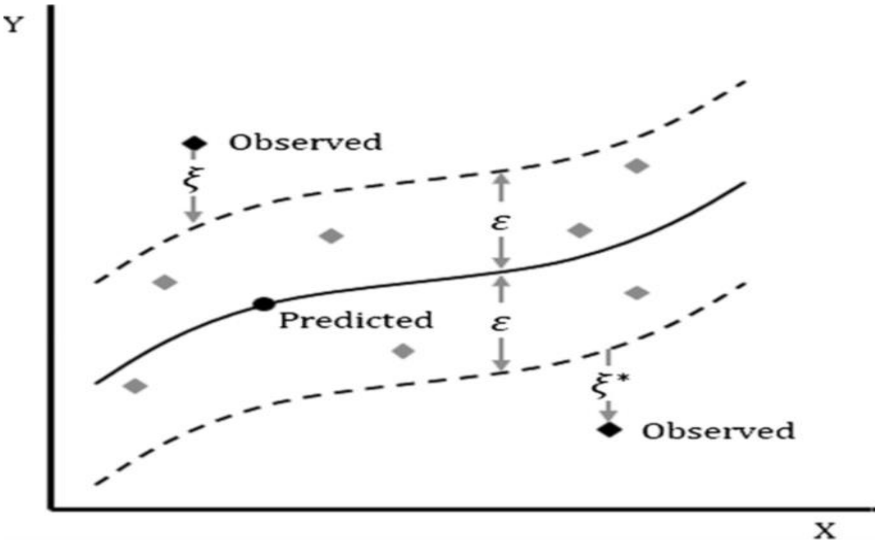


Figure (2.9): Non-linear SVR with Epsilon Intensive Band [46].

The relationship between inputs x_1; x_2; ….; x_n and output Y is determined as equation (2.16) [45]:

$$Y = W\varphi\ (x) + b \tag{2.16}$$

where x is the input parameter; φ (x) represents the high-dimensional feature spaces (kernel function), which is nonlinearly mapped from the input space x; ω is the weight coefficient; b is the deviation value. The coefficients of ω and b are estimated by minimizing the regularized risk function as equation (2.17) [47]:

$$\frac{1}{2}\|\omega\|^2 + C\frac{1}{n}\sum_{i=1}^{n} L_\varepsilon(y_i, f(x_i)) \tag{2.17}$$

where $||\omega||^2$ is a regularized term; minimizing the regularized term can make a function as flat as possible. C is the regularization constant (the cost of making an error) and ε is the threshold of the support vector machine. The term (1/n) $\sum_{i=1}^{N} L_\varepsilon(\mathrm{yi}, \mathrm{f(xi)})$ is the empirical error measured by the ε-insensitive loss function, as expressed in equation (2.18) [47]:

$$L_\varepsilon\big(y_i, f(x_i)\big) = \begin{cases} |y_i - f(x_i)| - \varepsilon, |y_i - f(x_i)| \geq \varepsilon \\ 0 \end{cases} \tag{2.18}$$

Here determines a ε-Support Vector Regression (ε-SVR). If the predicted value is within the tube, the loss value is zero, while if the predicted value is outside the tube, the loss value is magnitude of the difference between the predicted value and the radius ε of the tube. To get the estimation of ω and b, the equation (2.16) is transformed to the primal objective function (2.19) [45] by introducing the positive slack variables ξ_i^*.

$$\min\frac{1}{2}\|w\|^2 + \mathrm{C}\frac{1}{N}\sum_{i=1}^{N} (\xi_i + \xi_i^*) \tag{2. 19}$$

with the following constraints:

$$Y_i - W\varphi(x_i) - b \leq \varepsilon + \xi_i$$

$$W\varphi(x_i) + b - Y_i \leq \varepsilon + \xi_i^*$$

$$\xi_i , \xi_i^* \geq 0$$

ξ_i and ξ_i^* are the residuals beyond the ε boundary [45,48], presented in figure (2.9).

The selection of kernel function φ (x) is the crucial to the accuracy of the final prediction model. Due to the less numerical difficulties and nonlinear property in a high dimensional space, radial basis function (RBF) is chosen as the kernel function, which is denoted as equation (2.20) [45]:

$$k(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \qquad (2.20)$$

where $\|x - y\|^2$is the squared Euclidean distance between the two feature vectors, and σ is a kernel parameter.

In general, the parameters of c and σ are significant factors that directly impact the accuracy of the prediction model.

2.12 Naive Bayesian

It is one of the classification algorithms and depends on the Bayes theory.

2.12.1 Bayes' Theorem

A theory that calculates certain probabilities is conditional since the probability that reflects the impact of an event on the likelihood of another event is known as conditional probabilities. Bayes' theory uses the posterior probability and the previous probability. It represents the pre-probability of an event or hypothesis of the original probability where it was obtained before obtaining any additional information. The revised probability of the event through the use of additional information or evidence that were obtained is known as the posterior probability [49].

The theory is written as equation (2.21) [49]:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \qquad (2.21)$$

Where,

The prior probability of A is P(A)

The prior probability of B is P(B)

The posterior probability of A given B is P(A|B)

The posterior probability of B given A is P(B|A)

2.12.2 Naive Bayesian Classifier

The Naive Bayes classifier is a simple and convenient probabilistic classifier that depends on the application of the Bayes theorem. Naive Bayes regards each component of the attributes as an independent variable. This classifier can be trained in supervised learning very well, and can also be used in complicated real-life situations.

All features of the training examples assumed independent from one another. The Naive Bayes classifier represents every pattern (X) as a vector that has n dimensions for attribute values represent by [a1, a2, a3, ...an]and that there are class's [c1, c2, c3....cn]. As equation (2.22) [49], X is assigned to class if and only if

$$P(C_i|X) > P(C_J|X) \qquad (2.22)$$

$For\ 1 <= j < i\ and\ j \neq i$ using equation (2.21), getting the equation (2.23) [49]:

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \qquad (2.23)$$

the classifier makes the naive assumption that the features (Which is indicated by n for its total number) are conditionally independent of one another to reduce the computational expenses to approximate the probability of continuous data set, the equation (2.24) [49] can be used:

$$P(X|C_i) = \prod_{j=1}^{n} P(x_j|C_i) \qquad (2.24)$$

Where $P\left(C_J\right) = \frac{|C_i|}{N}$ and P(X)=constant for each class, needs Naive Bayesian classifier to increase $P(X|C_i)$ only, because it calculates class distribution only and this leads to a reduction in the cost of calculation.

Bayesian classifier only requires one data scanning so it is very simple and provides high accuracy [49].

2.12.3 Gaussian Naive Bayes

A common assumption when working with continuing data is that the continuous values that correspond with each class are allocated to the Gaussian distribution.

Training results are broken down by classes as equation (2.25) [50]:

$$P(C_i) = frequency\ (C_i)/N \qquad (2.25)$$

Where, N is the total number for records and C_i is the class number.

The mean and standard deviation is measured for each class according to equation (2.26) [50] and equation (2.27) [50]:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad (2.26)$$

$$\sigma = [\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2]^{0\cdot 5} \qquad (2.27)$$

Where

n is the number of instances.

x_i is a specific value of the x variable for the ith instance.

Therefore, the equation (2.28) [50] probability distance function (pdf) can be used to approximate the probability of continuous data set [50].

$$P(X = x|C = c) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \qquad (2.28)$$

In order to understand how Naïve Bayes works, this simple example is explained, so that there are four features and a problem with two categories, as shown in table (2.1), and there is a need to apply the Naïve Bayes classifier.

Table (2.1) Sample Data for Naïve Bayes Classifier

Attribute				class
A1	A2	A3	A4	
2	1	4	3	C2
1	3	2	2	C1
4	2	1	2	C2
3	3	1	3	C2

To classification a new instance X= [3, 2,1,1],

Solution as follows:

Trian phase:

1- The probability is calculated for each class according to the equation (2.25).

P (C1) = frequency (C1)/ N = 1/4= 0.25

P (C2) = frequency (C2)/ N = 3/4=0.75

2- the mean and standard deviation are calculated according to equation (2.26) and equation (2.27) for every attribute in dataset of each class. as follows:

μ(A1) =1/1=1

σ(A1) =1/2 √ [$(2-1)^2$+$(1-1)^2$+ $(4-1)^2$+$(3-1)^2$]=1.9

As shown in table (2.2):

Table (2.2): Mean and Standard Deviation

mean							
Class 1				Class 2			
A1	A2	A3	A4	A1	A2	A3	A4
1	3	2	2	2.3	2	2	2.7
Stander deviation							
Class 1				Class 2			
A1	A2	A3	A4	A1	A2	A3	A4
1.9	1.1	1.2	0.7	1.5	0.9	1.2	0.6

Test phase

The Gaussian distribution is calculated for unlabeled testing instance according to equation (2.28).

P (A1=1|C1) =$1/\sqrt{2\pi}1.9e^{\frac{-(3-1)^2}{2(1.9)^2}}$

$=1/2.5*1.9e^{\frac{-4}{7.2}}$=0.4332

As shown in table (2.3):

Table (2.3): Probability Density Function

P(X|C1)				P(X|C2)			
3	2	1	1	3	2	1	1
0.4	0.3	0.34	0.1	0.24	0.5	0.2	0.1

1- Posterior probability of X is calculated according to equation (2.29)[50]:

$$P(C_j|X) = P(X|C_j)P(C_j) \tag{2.29}$$

P (C1|X) = (0.4*0.3*0.34*0.1) *0.25

=0.00306

P (C2|X) = (0.24*0.5*0.2*0.1) *0.75

= 0.0018

So, P(C1|X) is larger than P (C2|X), then pattern X is expected to be in class C1.

2.13 Accuracy Metrics

Several requirements are being used to evaluate various algorithms for the performance. Root mean square error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE) performance indices were used for comparison of target values and actual values in the performance measurement layer of the current proposal. The RMSE calculates the discrepancy between the energy expected and the power intended, the MSE is a calculation used to minimize the distribution of errors. And the MAPE is a metric that measures the difference in predictions as a percentage of the target capacity. The performance of RMSE, MAE, and MAPE can be calculated in equations (2.30)– (2.32) [51], as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{n} |T_i - P_i| \tag{2.30}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{n} \frac{|T_i - P_i|}{T_i} \times 100 \tag{2.31}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(T_i - P_i)^2} \tag{2.32}$$

Where N indicates the values as a whole, T represents the target value and P denotes the value expected. Such measures have a common standard for calculating the precision of the tests of the various algorithms [51].

Chapter Three

The Proposed Model

Chapter Three

The Proposed Model

3.1 Introduction

This chapter address the problem of electricity consumption prediction by testing different algorithms LSTM, KNN, SVR and Naive Bayes. And then the proposed will be that produced the best result among them. It will start by presenting the general proposed system.

3.2 The General Proposed System

In this thesis four systems were design modeled, and tested in order to find the best technique for such data. four algorithms LSTM, KNN, SVR and Navies Byes were tested, at different prediction time: short term prediction (one day, one week) and mid-term prediction (one month).

In the conducted experiment basic stages have been implemented. The first stage is read dataset, the second stage is the pre-processing, the third stage is building prediction model, and finally the last stage is the performance evaluation. As shown in the figure (3.1):

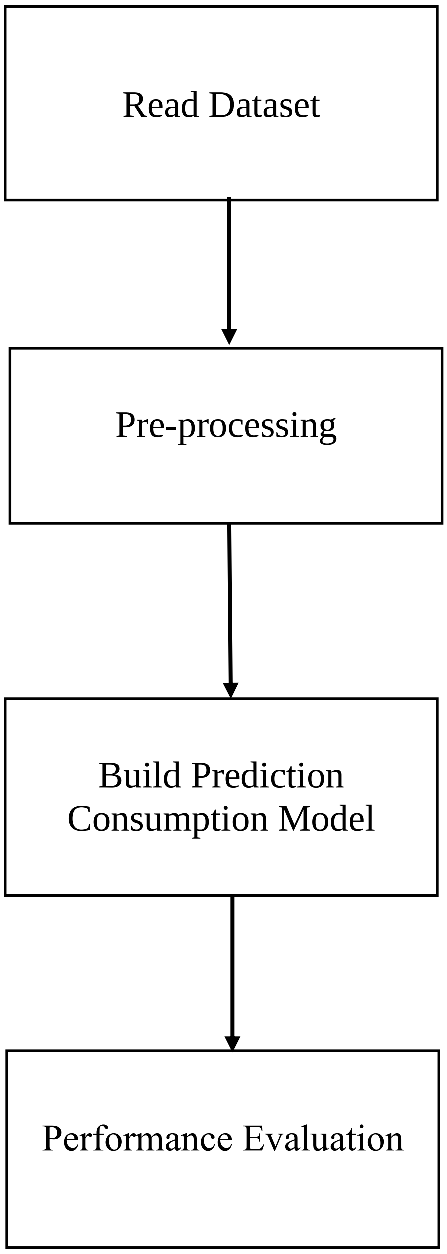


Figure (3.1): Block Diagram of General Proposed System

3.2.1 Dataset

In order the method for predicted electricity consumption for the proposed system. A dataset was collected from the source for different categories. As mentioned before, in this system four techniques were used LSTM, KNN, SVR and Navies bays. The predication system was implemented on dataset, gained from 2075259 measurements gathered in a house located in Sceaux (7km of Paris, France) between December 2006 and November 2010 (47 months).

3.2.2 Data Pre-processing

The preprocessing of input data is a very important part of the modeling process. It involves checking the quality of the input data and ultimately improving the types of inputs, chosen steps and time frames. It directly affects prediction results, their accuracy and reliability. Data pre-processing relies on cleaning the data. The data cleaning step is mandatory to remove all poor-quality information like missing data.

Algorithm (3.1): **Preprocessing Algorithm**

```
Input: data set
Output: data set after prepressing
Begin
Step 1: read data set.
Step 2: identify the type of each parameter.
Step 3: merging two columns date and time.
Step 4: dealing with missing Values, take the mean for each column according to the equation (2.26) and put it in the empty place for this column.
Step 5: normalize the dataset by apply min-max normalization according to the equation (2.1)
END
```

3.2.3 Prediction Models

This stage includes the prediction models that have been tested for the proposed system. Before choosing the proposed system, experiment has been conducted on LSTM, KNN, SVR and Naive Bayes. The details for each system are describe as follows:

3.2.3.1 LSTM Algorithm

LSTM (Long Short-Term Memory) Networks are called fancy recurrent neural networks.

LSTM can hold information for long periods of time due to its chain like structure, where it can solve the tasks that are difficult to implement using traditional RNN.

LSTM consists of three main parts:

Forget gate— there is information that are no longer needed to complete the task. This gate removes it, and this improves the performance of the network.

Input gate— through this portal, information is added to cells.

Output gate— this portal produces the necessary information.

The LSTM cell is created:

From the input layer, the previous hidden cell Ht-1 is entered and the new sequence x_t is entered, where the first step of this combined entry is that it is crushed through the tanh layer where tanh takes large or small variable numbers and converts them at a specific rate between (-1,1), in order to ensure that the numbers are in homogeneity and unevenness to generate a new memory C`(t) according to equation (2.9).

As for the input gate is a layer of sigmoid activation nodes, whose output is multiplied by the output of the tanh, the sigmoid of this input gate can stop any element of the input vector not required as this function outputs the values between 0 and 1, according to eq. (2.7).

And in forget gate is making an assessment on whether the past memory cell is useful for the computation of the current memory cell. Thus, the forget gate looks at the input and the past hidden state. This addition process, instead of multiplication, helps reduce the risk of gradient vanishing. This gate helps the network know the status variables that must be remembered or forgotten according to eq. (2.6).

After completing the above parts, the state of LSTM is updated according to eq. (2.10). Obviously, this equation connects the pre-state C_{t-1} and the present temporary-state C'_{t}.

Through the output gate, LSTM outputs the specified state, based on the cell status, where runs a sigmoid layer to determine the unit state section to be exported according to eq. (2.8). And deals current output o_t and state Ct with a tanh layer to write a new hidden layer state h_t according to eq. (2.11).

During the training process, the weight matrices W_f, W_i, W_o and W_c and bias vectors are learned by using Adam optimization.

As shown in algorithm (3.2).

Algorithm (3.2): LSTM cell Network Algorithm
Input: X (current input) H (previous hidden state) C (previous memory state) Output: H (current hidden state) C (current memory state)

```
Begin
Training phase
 step (1): training = {($x_t$, $x_{t+1}$) t = 1, 2, · · ·, T1} and validation set X
validation = {($x_t$, $x_{t+1}$), t = T1 + 1, T1 + 2, · · ·, T2},
Step (2): initialize W randomly, ∓ val
Step (3): adjusting W:
Step (4): for epoch = 1 to 200 do
 Perform forward propagation recurrently using equation from (2.6) to (2.11) as follows:
Step (4.1): the past memory state $C_t - 1$ is taken by the LSTM cell and it performs a wise multiplication of the element with the forgot gate (f).
                    $C_t = C_t - 1 * f_t$
(f gate gives values 0 or 1)
        if f   = 0 then past memory state is fully forgotten
        if f = 1 then past memory state passed to the cell H
step (4.2): compute new memory state from input state and $C'_t$ layer, with present memory state $C_t$
   $C_t$= $C_t$ + ($I_t$* $C`_t$)
   $C_t$ = present memory state at time step ‘t’, and it gets push through to next time.
Step (4.3): apply Tanh to $C_t$ then we do element wise multiplication with the output gate O, that will be our current hidden state $H_t$
 $H_t$ = Tanh $C_t$
pass Ct and $H_t$ to next time step, then repeat the process itself.
Step (4.4): compute output error: $xe_{t+1} - x_{t+1}$, t = 1, 2, · · ·, T1
Step (4.5): if error ≥ 0.001 then perform backward propagation for all layer n-1 to layer 1.
Step (4.6): update W
```

```
 $W = W + \Delta W$
step (4.7): perform forward propagation recurrently to update the network states.
validation phase
Step (4.8): read validation data.
step (4.9): perform forward propagation recurrently to compute $Xe = \{xe_{t+1}, t = T1 + 1, T1 + 2, \cdot\cdot\cdot, T2\}$
step (4.10): save the current W.
if epoch ≥ 200 then
break
test phase
step (5):  read test data
step (6): perform forward propagation recurrently to compute Xe.
Step (7): calculate RMSE, MAE, and MAPE.
End
```

3.2.3.2 KNN Regression Algorithm

The algorithm is used for regression, which is a non-parametric method. The closest examples of training k in the feature field are those that represent input. Through a similar calculation, the object property is obtained and from the average value of the closest training points, the value of the object is obtained. The flow chart of the KNN algorithm as shown in figure (3.2). KNN algorithm is clarified in algorithm (3.3).

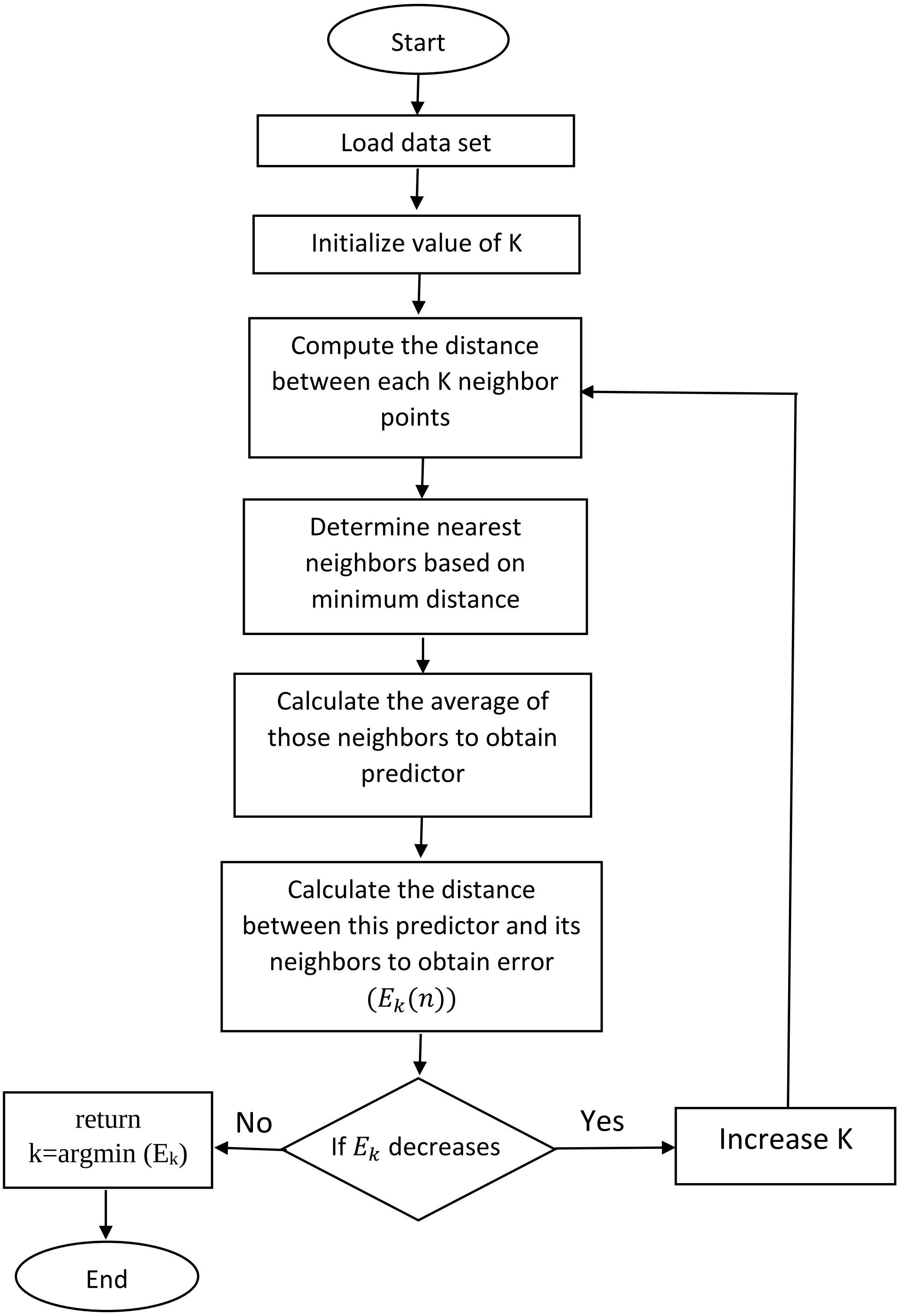

Start
Load data set
Initialize value of K
Compute the distance between each K neighbor points
Determine nearest neighbors based on minimum distance
Calculate the average of those neighbors to obtain predictor
Calculate the distance between this predictor and its neighbors to obtain error $(E_k(n))$
If E_k decreases
Yes
Increase K
No
return k=argmin (E_k)
End

Figure (3.2) KNN Regression Flowchart

Algorithm (3.3): KNN Algorithm

```
Input: features (Global active power, Global reactive, power, Voltage, Global intensity, Sub metering 1, Sub metering 2, Sub metering 3)
Output: predicated samples
Begin
Training Phase
Step (1): read in data and needed packages
Step (2): initialize value of K.
Step (3): calculate distances between each k neighbor points.
Distances [Index] = distance function in equation (2.14).
Step (4): determine nearest neighbors based on minimum distance.
Step (5): calculate the average of those neighbors to obtain predictor according to eq. (2.15).
Step (6): for n=0.
Step (7): calculate the distance between the predictor and its neighbor to obtain error($E_k(n)$).
Step (8): n=n+1.
Step (9): if $E_k$ decrease increase k and go step 3
Else stop and return k=argmin (Ek).
Testing phase
Step (10): read testing data.
Step (11): calculate the average of k neighbors (that was calculated in training).
Step (12): measure the distance between the test data and the average K neighbors.
Step (13): calculate RMSE, MAE, and MAPE.
 End
```

3.2.3.3 SVR Algorithm

Finding a function f(x) that has at most deviations from the actual goals of (y_i) of the available training data, is what SVR algorithm represents. Finding suitable values for excessive parameters through multiple rounds of model building is the main test part of the SVR.

There are training parameters (C, kernel, and ε) that must be prepared for ε - sensitive loss function, and this is what SVR depends on in its performance. The value of (C and ε) for any type of kernel affects the difficulty of the model. Also, the number of support vectors used for prediction is affected by the value of (ε), as the larger value of (ε) the less number support vectors. This makes regression estimates less complicated. Figure (3.3) shows the flow chart of SVR algorithm. SVR algorithm is clarified in algorithm (3.4).

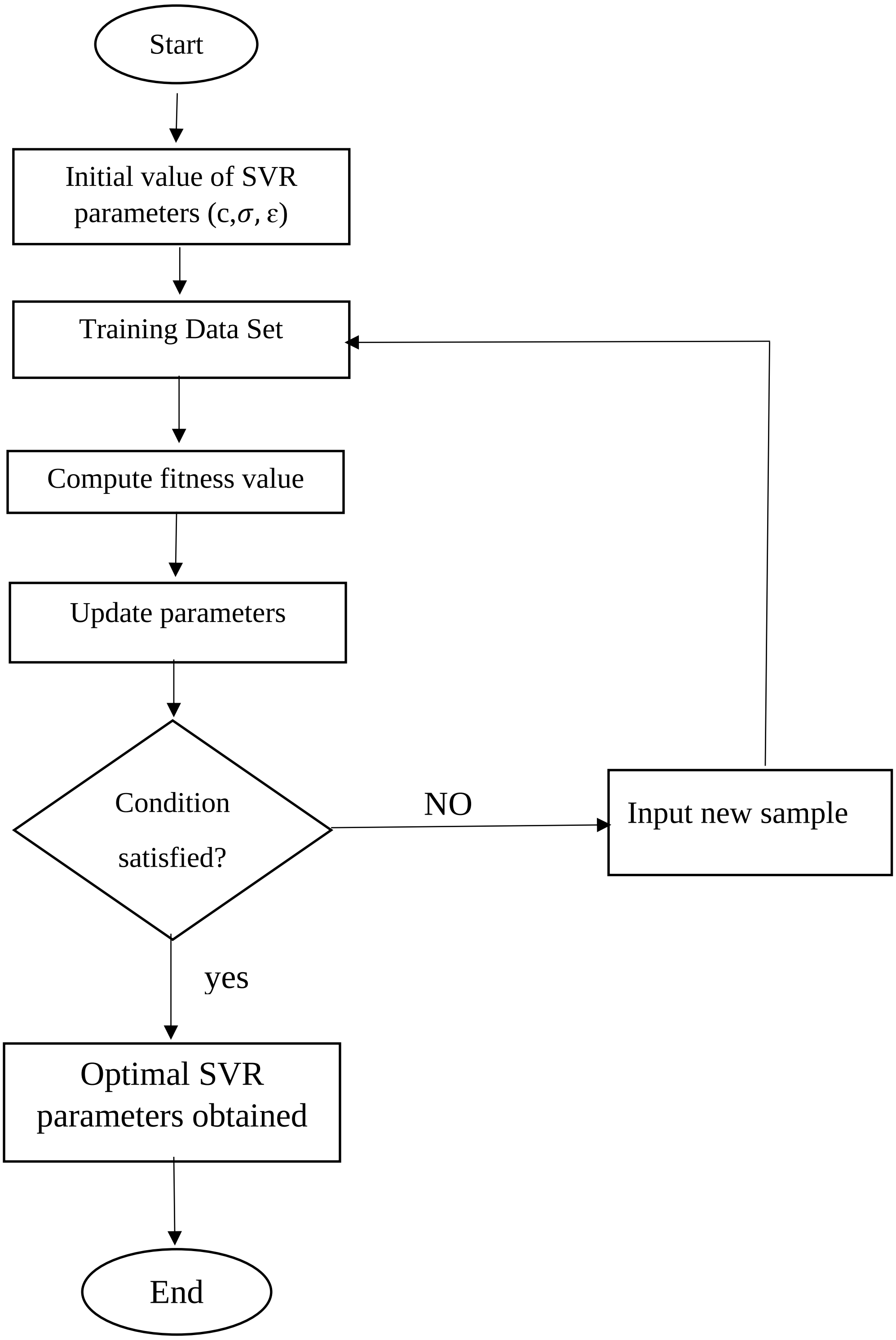


Figure (3.3) SVR Flowchart

Algorithm (3.4): SVR Algorithm

```
Input: features (Global active power, Global reactive, power, Voltage, Global intensity, Sub metering 1, Sub metering 2, Sub metering 3)
Output: predicated samples

Begin
Training Phase
Step (1): reading the dataset.
Step (2): set an initial value of weight and bias.
Step (3): set an initial value of parameter C and kernel parameter $\sigma$.
Step (4): set $\varepsilon$ to 0.0001.
Step (5): calculate the kernel RBF according to eq. (2.16) and eq. (2.20).
Step (6): obtain the best value of C according to eq. (2.19).
Step (7): use the best value of C on the training data.
Step (8): update the values of weight and Bias.
Step (9): if the difference between the update values of weight and Bias and the previous value is greater than $\varepsilon$ go to step 5
Else end training.
Testing Phase
Step (10): select the testing data.
Step (11): implement the test by using the best value of weight, bias, C and $\sigma$ from the training.
Step (12): calculate RMSE, MAE and MAPE.
END
```

3.2.3.4 Naive Bayes Algorithm

The last algorithm that used is Naive Bayes, and the flow chart as shown in figure (3.4). Naive Bayes algorithm is clarified in algorithm (3.5).

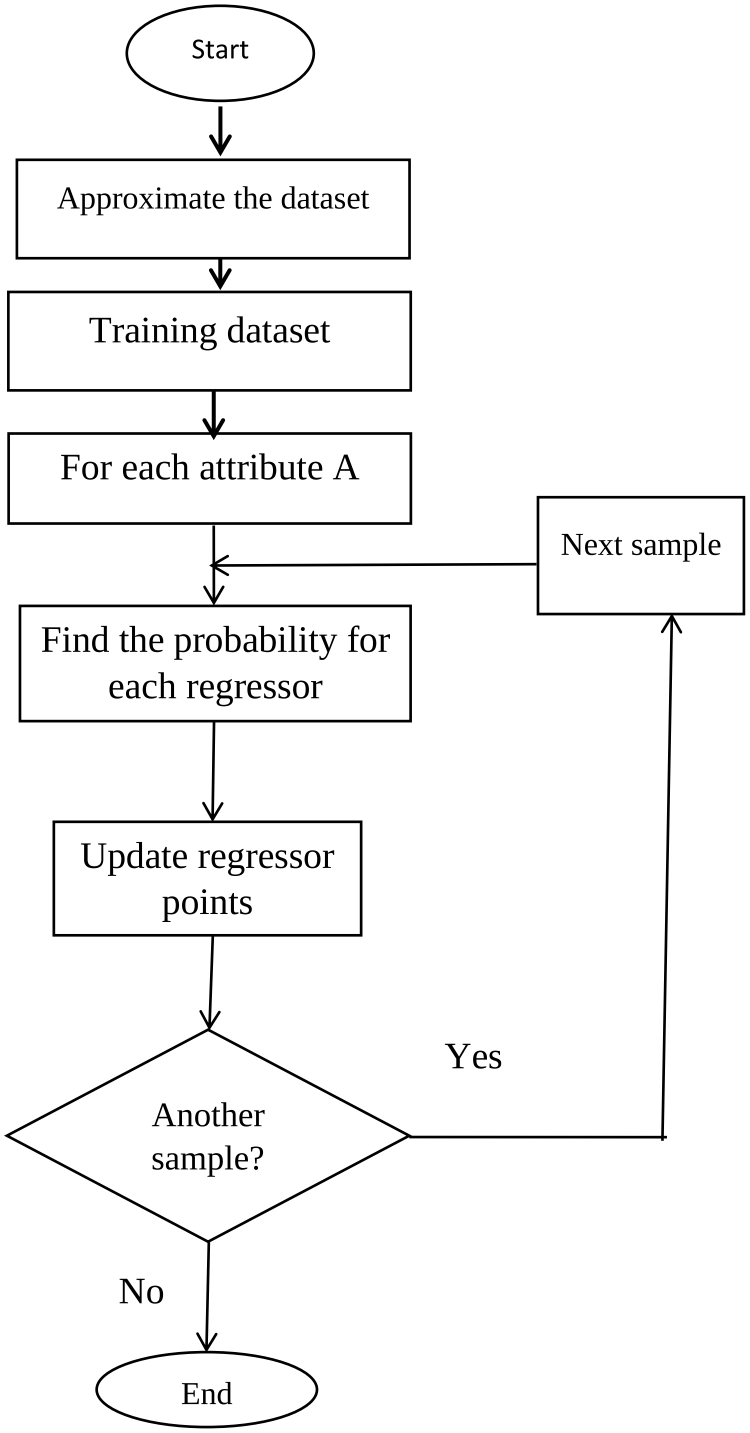


Figure (3.4) Naive Bayes Flowchart

```
Algorithm (3.5): Naive Bayes Algorithm

Input: attributes (Global active power, Global reactive, power, Voltage, Global intensity, Sub metering 1, Sub metering 2, Sub metering 3)
Output: regressor label.

Begin
Step (1): Approximate the dataset.
          Splitting the dataset into 70% train data and 30%test data.
Training Phase
Step (2): Total=All instances in training dataset
Step (3): Calculating the probability of each regressor
P(Cj)=frequency (Cj) / Total
Cj is regressor in training dataset.
j is number of regressor in training dataset.
Step (4): Calculating the mean and standard deviation values for each attribute of each regressor in training dataset by applying the equation (2.26) and equation (2.27).
Testing Phase
Step (5): X is an instance in testing dataset
Step (6): Calculating the Probability Density Function(pdf) of X at Cj, for values of attributes of X exists in S, p (Xi | Cj) by applying the equation (2.28).
Step (7): Calculating conditional probability of X at Cj for values result from step (3.2), using equation (2.24).
Step (8): Calculating posterior probability of X, p (Cj|X) that represent probability of instance at Cj using equation (2.29).
P (Cj | X) = P(X | c j) p(C j) // probability of instance at $C_j$
Step (9): Assign regressor label to the test sample X based on maximum posterior p(Cj|X).
Step (10): Return regressor label.
Step (11):  calculate (RMSE, MAE, and MAPE).
END
```

3.2.4 Performance Evaluation

To evaluate the algorithms for the performance., mean absolute error (MAE) according to equation (2.30). mean absolute percentage error (MAPE) according to equation (2.31) and Root mean square error (RMSE) according to equation (2.32) performance indices were used for comparison of target values and actual values in the performance measurement layer of the algorithms. These scales provide a single value for measuring the accuracy of results for different algorithms.

3.3 The Prediction System

This section includes the systems that have been used in the thesis. Testing the approaches with three prediction time, short term prediction (one day, one week), mid-term prediction (one month). the dataset is divided into 70% training and 30% testing, the flowchart of the proposed system is shown in figure (3.5), and Algorithm (3.6).

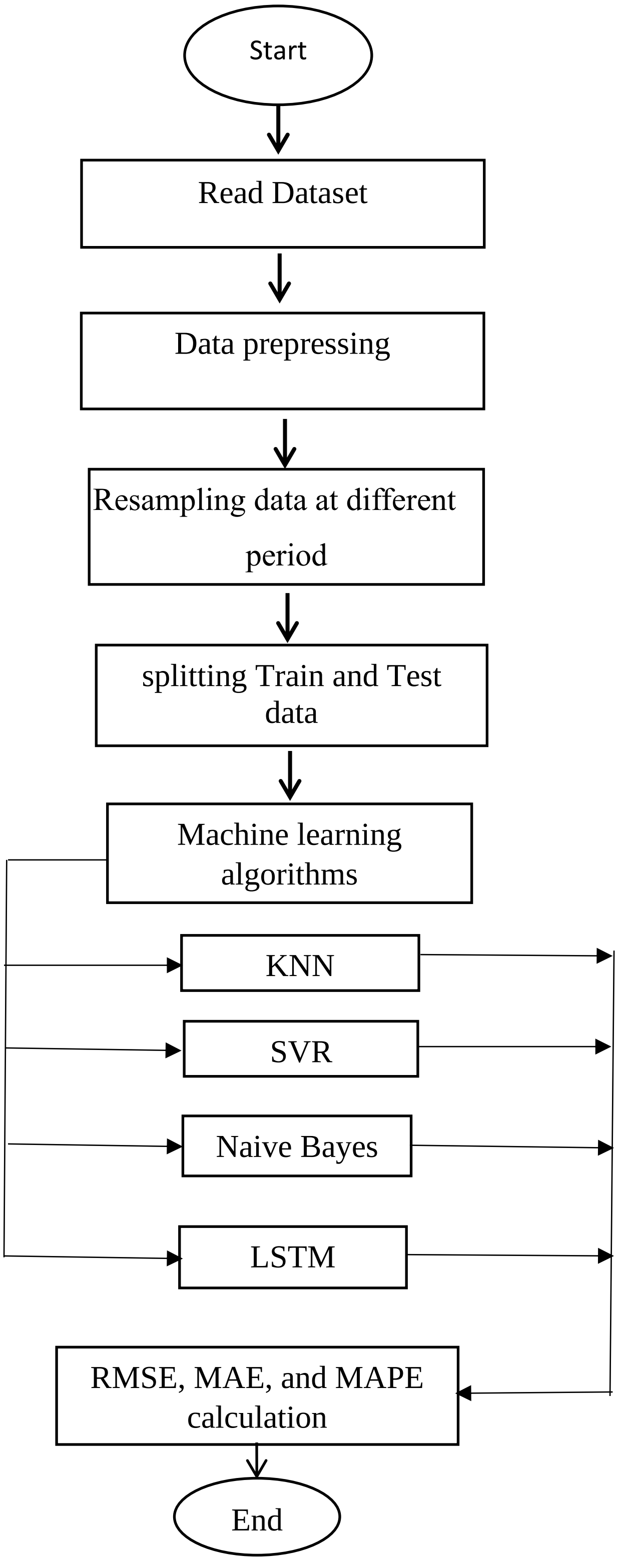

Start
Read Dataset
Data prepressing
Resampling data at different period
splitting Train and Test data
Machine learning algorithms
KNN
SVR
Naive Bayes
LSTM
RMSE, MAE, and MAPE calculation
End

Figure (3.5) The Proposed Prediction System Flowchart

Algorithm (3.6): The Proposed Prediction System Algorithm

```
Input: data set.
Output: proposed prediction system
Begin
Step (1): read Data.
Step (2): define each parameter.
Step (3): make a preprocessing.
Step (4): Resampling data at different period (one day, one week, on month).
Step (5): splitting Train and Test data.
Step (6): build a prediction model using LSTM, KNN, SVR and Naïve Bayes.
Step (7): calculate RMSE, MAE, and MAPE.
Step (8): make a comparison between the algorithms.
END
```

Chapter Four

Experimental Results and Evaluation

Chapter Four

Experimental Results and Evaluation

4.1 Introduction

In this chapter, the implementation results obtained by testing different prediction models are summarized and then the proposed model will be that which produced the best result among them described in detail in chapter three. The next sections discuss the experimental results which obtained from the preprocessing and prediction, which contain training and testing stages. Finally, this chapter includes the comparison of the proposed work with other existing works.

4.2 Implementation Environment

Electricity consumption predicting approaches using LSTM, KNN, SVR and Naïve Bayes is implemented under a specific system requirement such as Windows-10 operating system, Hardware processor: Core i5- CPU 8250U, 1.60 GHz, and (8GB) RAM. Python 2018 (3.8 64-bit) programming language with TensorFlow backend.

4.3 Prediction Systems Implementation

The prediction systems have four stages executed sequentially, starts by read dataset ends with performance evaluation as described in the following sections.

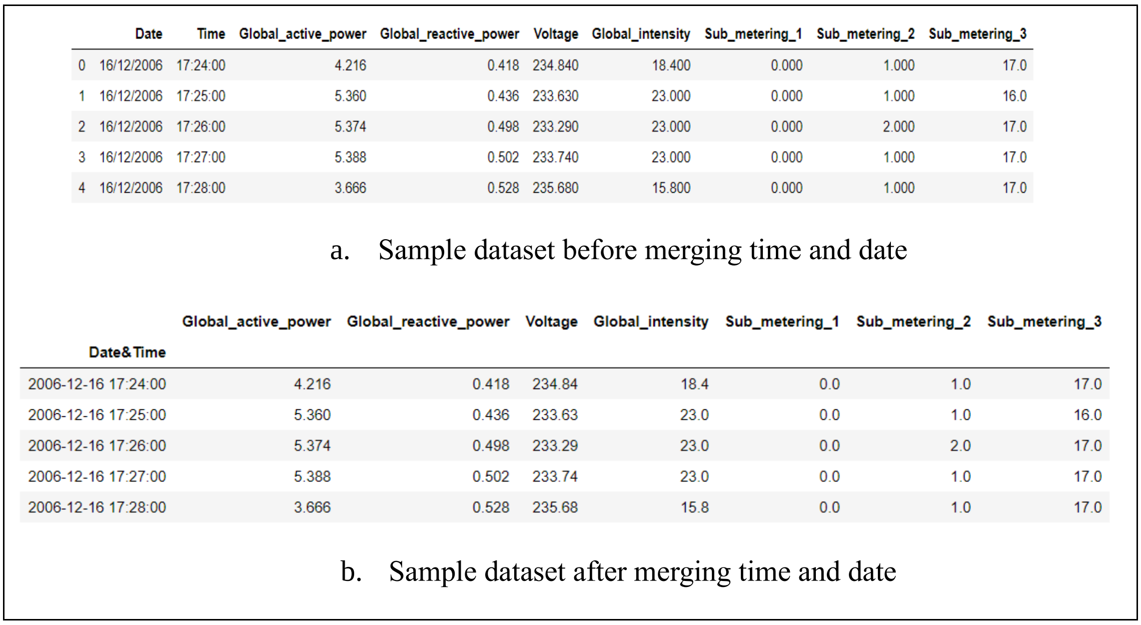
4.3.1 Dataset Information

Electricity energy consumption dataset [52] contains 2075259 measurements. The consumption of power in period of four years (between December 2006 and November 2010 (47 months)) for one house located in Sceaux ((7km of Paris, France)) was delineated using a dataset of a multivariate time series, which consist in addition to the date and time seven attributes are described in the appendix A.

Where this dataset was used because it represents globally approved and accurate data and contains other features in addition to the feature of actual consumption and affects it, through which the forecast of electricity consumption can be measured compared to other data that depend only on the characteristic of actual electricity consumption.

4.3.2 Pre-processing

The preprocessing of the entered data is a very important part of the modeling process, as it directly affects prediction results, their accuracy and reliability, and this section highlights the data cleaning process used in experiments. the input data is read and the type of each parameter are determined, and because the data is large in size as it calculate consumption per minute, the columns of time and date are combined to ensure the ease of dealing with data as shows in figure (4.1).



	Date	Time	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
0	16/12/2006	17:24:00	4.216	0.418	234.840	18.400	0.000	1.000	17.0
1	16/12/2006	17:25:00	5.360	0.436	233.630	23.000	0.000	1.000	16.0
2	16/12/2006	17:26:00	5.374	0.498	233.290	23.000	0.000	2.000	17.0
3	16/12/2006	17:27:00	5.388	0.502	233.740	23.000	0.000	1.000	17.0
4	16/12/2006	17:28:00	3.666	0.528	235.680	15.800	0.000	1.000	17.0

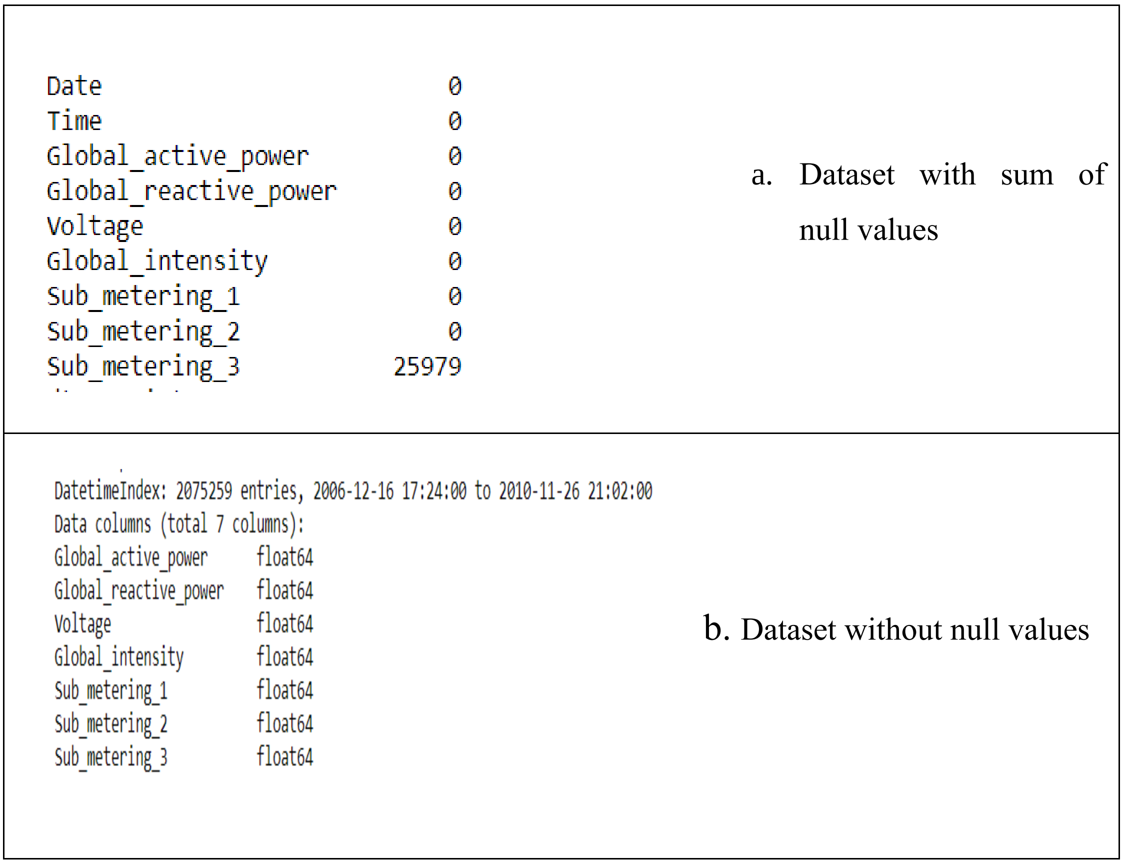
a. Sample dataset before merging time and date

	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
Date&Time							
2006-12-16 17:24:00	4.216	0.418	234.84	18.4	0.0	1.0	17.0
2006-12-16 17:25:00	5.360	0.436	233.63	23.0	0.0	1.0	16.0
2006-12-16 17:26:00	5.374	0.498	233.29	23.0	0.0	2.0	17.0
2006-12-16 17:27:00	5.388	0.502	233.74	23.0	0.0	1.0	17.0
2006-12-16 17:28:00	3.666	0.528	235.68	15.8	0.0	1.0	17.0

b. Sample dataset after merging time and date

Figure (4.1) Sample Dataset before and after Merging Time and Date

Each row was examined with a date versus each column to determine any null values. If null values are found, this is replaced by taking the average value according to equation (2.26) for all other values with a similar time period in the data set for this respective column.



```
Date                         0
Time                         0
Global_active_power          0
Global_reactive_power        0
Voltage                      0
Global_intensity             0
Sub_metering_1               0
Sub_metering_2               0
Sub_metering_3           25979
```

a. Dataset with sum of null values

```
DatetimeIndex: 2075259 entries, 2006-12-16 17:24:00 to 2010-11-26 21:02:00
Data columns (total 7 columns):
Global_active_power      float64
Global_reactive_power    float64
Voltage                  float64
Global_intensity         float64
Sub_metering_1           float64
Sub_metering_2           float64
Sub_metering_3           float64
```

b. Dataset without null values

Figure (4.2) Dataset with and Without Null Values

In the figure (4.2a) it shows the sum of the null values for each column, where it shows that the column of “sub metering 3” contains 25979 of the null values while the other columns contain zero of the null values. The figure (4.2b) shows that all columns contain values and their type is float. This shows that we got rid of the null values in the data.

Data can be pre-processed by using the Min-Max Scaler method, where the features are transformed by changing the size of each feature to a specific range were normalized to fit them in the interval [0, 1] by using equation (2.1) where the maximum and input feature range values are min and max, and each feature is measured separately so that it is in the points specified in the training set.

4.4 Prediction Systems Results

This section will display the results obtained from the implementation of the prediction systems. It includes the results of the prediction of four systems, LSTM, KNN, SVR and the fourth Navies Bayes. Data are split into two sets: training and testing datasets. The training dataset is used to calculate the model parameters, while the testing dataset is utilized to measure the model’s performance.

In this study, after preprocessing for dataset, the dataset is divided into 70% training and 30% testing, where training and test data are divided into three different ways to be able to understand power consumption for individual house.

•Given one day predicting the next day

The model is trained to take one day of inputs to predict consumption for the next day.

•Given one week predicting the next week.

one week was given to predict one week ahead.

•Given one month predicting the next month.

The model is trained to take one month of inputs to predict consumption for the next month.

The prediction of one day and one week in the future represents a short-term prediction. As for the prediction of one month in the future, it is within the medium-range prediction. These ranges are used to create energy consumption behavior for 14 months (which represents 30% for data testing).

The input (X_t) is used to predict (X_{t+1}) as the output. (X_t) can represent one day, one week, or one month of power consumption for an individual house.

Where the prediction is made by entering the attributes values for the time period (t) to predict the energy consumption for the time period (t+1).

Algorithms are evaluated using Accuracy metrics mean absolute error (MAE) according to equation (2.30), mean absolute percentage error (MAPE) according to equation (2.31) and root mean square error (RMSE) according to equation (2.32). The lower the error ratio, the greater the accuracy of the model, the results as follows:

4.4.1 First System LSTM Results

In this section, the implementation of LSTM network will be discussed. input dataset after preprocessing and output is prediction of active power.

Training phase: 70% from dataset are used for training. Firstly, initialize weight randomly. Applying equations from (2.6) to (2.11) to have a stable weight, the LSTM trained them for several epoch.

Test phase: 30% from dataset are used for testing. The active power is predicted by starting with the weight and learning rate from the training step and applying equations from (2.6) to (2.11) on the entered test values. Where RMSE, MAE and MAPE are used to performance evaluation. as shown in its algorithm (3.2).

LSTM contain 100 layers, and used for train 200 epochs with batch size 70 for training dataset. To obtain the most precise output prediction. Adam optimizer has been utilized as well as mean square error.

Our data is in 2 dimensions matrix consisting columns and corresponding rows. But input to a LSTM network is 3 dimensions matrix. The other dimension of this matrix is time steps.

Table (4.1) Model summary (LSTM)

Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(None, 100)	43200
dropout_1 (Dropout)	(None, 100)	0
dense_1 (Dense)	(None, 1)	101
Total params: 43,301 Trainable params: 43,301 Non-trainable params: 0		

Table (4.1) shows the Model summary (LSTM), which has 100 hidden layers used including a dropout layer before the last layer. It is a layer used in deep neural networks to enhance the training area 's performance and get rid of the negative Overfitting phenomenon. And the dense layer which is the last layer from which the output was obtained. The table also shows the total number of parameters in the layers and the number of parameters that have been trained.

- **Performance Evaluation For LSTM**

The performance of a system LSTM is measured by calculating (MAE) according to equation (2.30), (MAPE) according to equation (2.31) and (RMSE) according to equation (2.32), for each range and according to the following:

1.Short term prediction (one day)

It shows by predicting one day ahead, along 14 months Which represents the duration of the test.

Table (4.2) Predicted LSTM for Active Power (One Day) over 14 Months

Actual Active Power (KWh)	Predicted Active Power (KWh)
1.48313194	1.03650101
1.41342222	1.19516655
0.98463056	1.17046042
0.86565	1.03457302
1.13653472	0.95700433
0.99045694	1.07159525

. . . .	. . . .
0.62563194	1.30569124
1.41773333	0.87601756
1.09551111	1.26984924
1.24739444	1.22108569
mean absolute error (MAE) =	**0.183**
mean absolute percentage error (MAPE) =	**18.324**
root mean square error (RMSE)=	**0.244**

Table (4.2) shows the Active Power values for the short term (one day), and through the implementation of LSTM algorithm (3.2), prediction values (one day) for 14 months will appear, and from the values of this table (MAE), (MAPE) and (RMSE) can be calculated to measure the accuracy of the prediction.

The table (4.3) shows the prediction errors of LSTM for one day by calculate (MAE), (RMSE) and (MAPE).

Table (4.3) Prediction Errors of LSTM for One Day

Prediction Errors of LSTM for One Day
MAE=0.183
MAPE=18.324
RMSE=0.244

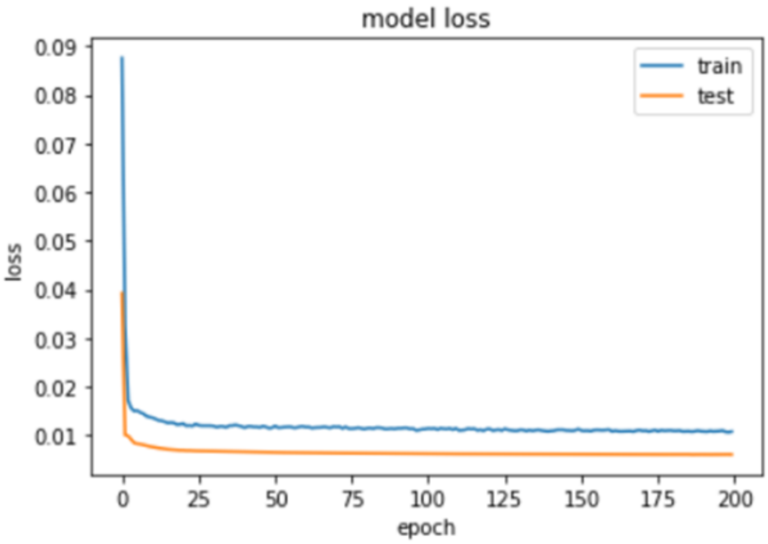


Figure (4.3) Model Loss for LSTM for One Day

For the short-term prediction (one day), as shows in figure (4.3), the loss is about 0.015 for the training and 0.010 for testing. It is clear that the error for the training is higher than for the testing.

2.Short term prediction (one week)

It shows by predicting one week ahead, along 14 months Which represents the duration of the test.

The table (4.4) shows the prediction errors of LSTM for one week by calculate (MAE), (RMSE) and (MAPE).

Table (4.4) Prediction Errors of LSTM for One Week

Prediction Errors of LSTM for one week
MAE=0.145
MAPE=15.182
RMSE=0.179

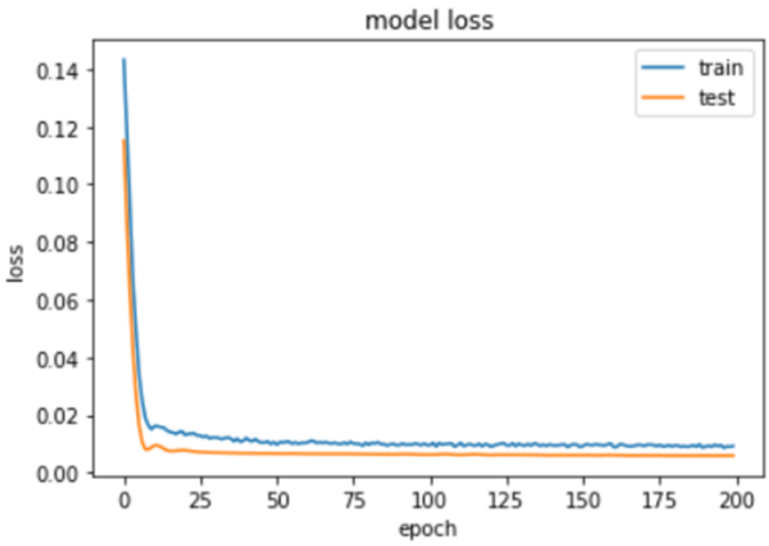


Figure (4.4) Model loss for LSTM for One Week

For the short-term prediction (one week), in figure (4.4). The loss is about 0.009 for the training and 0.006 for testing. It is noted that the loss value of the test tries to approach the training loss compared to short term prediction.

3.Medium term prediction (one month)

It shows by predicting one month ahead, along 14 months Which represents the duration of the test.

The table (4.5) shows the prediction errors of LSTM for one month by calculate (MAE), (RMSE) and (MAPE).

Table (4.5) Prediction Errors of LSTM for one month

Prediction Errors of LSTM for one Month
MAE=0.145
MAPE=14.018
RMSE=0.166

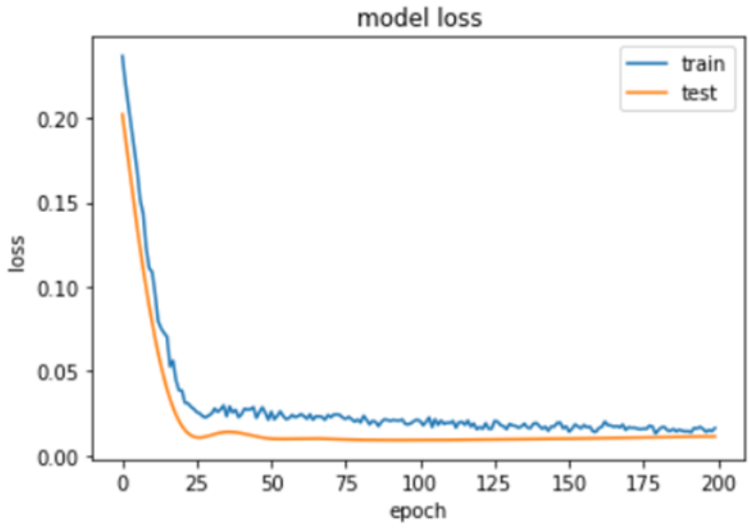


Figure (4.5) Model loss for LSTM for one month

For the mid-term prediction, in figure (4.5) shows that the loss is about 0.006 for the training and 0.004 for testing

By comparing the error scores for the periods for LSTM with each other, it can be seen that the RMSE, MAPE and RMSE value is low in mid-term predictions (one month) 0.145, 14.018 and 0.166. As show in table (4.5). This clearly indicates that LSTM has performed good with mid-term predictions.

4.4.2 Second System KNN Results

The input dataset after preprocessing and output is prediction of active power. The basic idea of KNN is as follow:

Training phase: 70% from dataset are used for training. A number of numerical features which consider the input and try to find the similarity between the input according to equation (2.14), determine nearest neighbors based on minimum distance. Then, calculate the average of those neighbors to obtain predictor according to equation (2.15), as discuss in chapter two. calculate the distance between the predictor and its neighbor to obtain error. If the error decreases then the value of k is increased, and return to calculate the distance between each neighbor's point, and if the error rises, we stop to get the point of k that got the least error.

Test phase: 30% from dataset are used for testing. The prediction for the active power is done by calculate the average of k neighbors (that was calculated in training). Then, measure the distance between the test data and the average K neighbors. Where RMSE, MAE and MAPE are used to performance evaluation. The results of KNN depending on (3.3) algorithm.

- **Performance Evaluation For KNN**

The performance of a system KNN is measured by calculating (MAE), (MAPE) and (RMSE) for Short term prediction (one day, one week) and medium-term prediction (one month). As shown in table (4.6).

Table (4.6) Prediction Errors of KNN

Prediction Errors of KNN			
	One day	for one week	for one month
MAE	0.228	0.170	0.170
MAPE	23.952	16.041	15.750
RMSE	0.300	0.213	0.198

By comparing the error scores for the periods for KNN with each other, it can be seen that the RMSE, MAPE and RMSE value is low in mid-term predictions (one month) 0.170, 15.750 and 0.198. As show in table

(4.6). This clearly indicates that KNN has performed good with mid-term predictions.

4.4.3 Third System SVR Results

There are slight differences in precept SVR action in relation to SVM classification, the input to the algorithm is dataset after preprocessing and output is prediction of active power. The data have been separated into training set and test set. The way to work is as follows:

Training phase:

70% from dataset are used for training. The hyper plane is been found according to eq.(2.16) and that by setting an initial weight and bias .The three training parameters are prepared (kernel, C, ε) where the Radial Basis Function (RBF) type kernel has been used, set ε to 0.0001 and vary C. Train the dataset, obtain the best value of C. Use the best value of C and σ on the training data according to eq.(2.19) and eq.(2.20). Update the values of weight and Bias.

The support vectors are affected by a value of ε. The complexity of the model depends on the values of (ε, C).

Testing phase:

30% from dataset are used for testing. Test data will estimate the prediction value using the train model parameters (σ, C, w) according to eq. (2.16), eq. (2.19) and (2.20), then the predicated values will be compere with the test data output for error calculations. Where RMSE, MAE and MAPE are used to performance evaluation. The results of SVR depending on (3.4) algorithm.

- **Performance Evaluation For SVR**

The performance of a system SVR is measured by calculating (MAE), (MAPE) and (RMSE) for Short term prediction (one day, one week) and medium-term prediction (one month). As shown in table (4.7).

Table (4.7) Prediction Errors of SVR

Prediction Errors of SVR			
	One day	One week	One month
MAE	0.191	0.147	0.160
MAPE	20.226	14.870	15.301
RMSE	0.250	0.179	0.176

By comparing the error scores of all prediction ranges for SVR with each other, and despite the convergence of the result RMSE for short term (one week) with the mid-term (one month), However, other results indicate that SVR excels in performance in short term (one week) 0.147,14.870 and 0.179 as show in table (4.7).

4.4.4 Fourth System Navies Bayes Results

Given that the data used are time series data, large size and floating type, it is difficult for us to use classification. And in order for the Navies Bayes algorithm to be used, Approximate the dataset to reduce the number of categories and facilitate the classification of data into these categories. The input to the algorithm is dataset after preprocessing and the result is the prediction of the active power. The data set was separated into a training group and a test group. The Navies Bayes algorithm has been implemented as follows:

Training phase: 70% from dataset are used for training. In this stage the mean and the standard deviation are calculated for attributes (Global active power, Global reactive power, Voltage, Global intensity, Sub metering1, Sub metering2 and Sub metering3) of the training data, according to equation (2.26) and equation (2.27) and then used as parameters in the testing phase.

Testing phase: 30% of the power consumption dataset has been used, to compute the posterior probability according to equation (2.29), the probabilities are predicted depending on the Gaussian distribution for each regressor according to equation (2.28) as discuss in chapter two. Where RMSE, MAE and MAPE are used to performance evaluation. as shown in its algorithm (3.5).

- **Performance Evaluation for Naïve Bayes**

The performance of a system Navies Bayes is measured by calculating (MAE), (MAPE) and (RMSE) for Short term prediction (one day, one week) and medium-term prediction (one month). As shown in table (4.8).

Table (4.8) Prediction Errors of Naive Bayes

	Prediction Errors of Naive Bayes		
	One day	One week	One month
MAE	0.303	0.199	0.171
MAPE	28.418	19.018	17.413
RMSE	0.426	0.254	0.216

By comparing the error scores for all prediction ranges for Naive Bayes with each other, it can see that the RMSE, MAPE and RMSE value are reduced in mid-term predictions (one month) as follows 0.171, 17.413 and 0.216 as show in table (4.8). This clearly indicates that Naïve Bayes has performed good with mid-term predictions.

4.5 Performance Comparison

In this section, four systems LSTM, KNN, SVR and Naive Bayes are tested to obtain the proposed system will be that produced the best result among them by comparing the results obtained from the accuracy measurements RMSE, MAPE and RMSE according to equations from (2.30) to (2.32) for short term (one day and one week), and mid-term predictions time. And time was calculated for training on a training dataset where time was measured in seconds.

4.5.1 Short Term Prediction (one day)

In order to show short term predictions (one day), the use of models to predict power consumption. In table (4.9) The results are shown. In comparison to KNN, SVR and Naïve Bayes, LSTM has shown to perform better for short term predictions (one day), although it had the most training time.

Table (4.9) Short Term Predictions (One Day)

model	Error			Time (second)
	MAE	MAPE	RMSE	
LSTM	0.183	18.324	0.244	17.59299
KNN	0.228	23.952	0.300	0.054105
SVR	0.191	20.226	0.250	0.100896
Naïve Bayes	0.303	28.418	0.426	0.001290

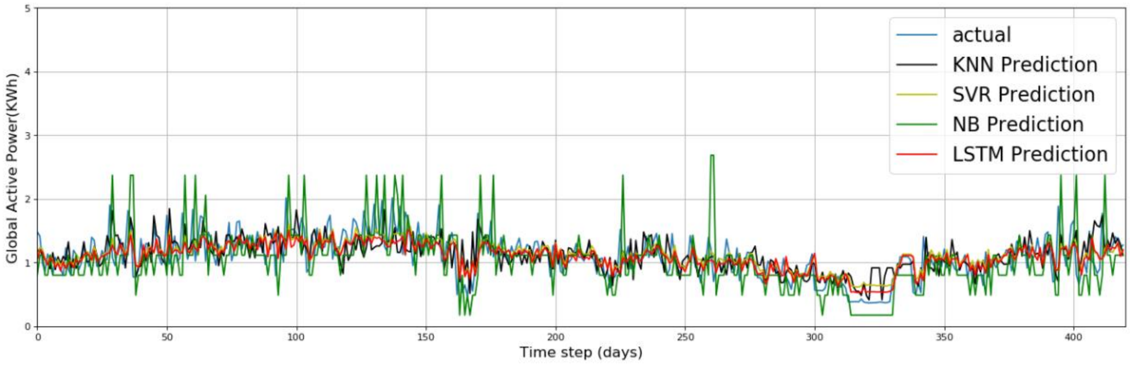


Figure (4.6) Actual Consumption vs. Predicted Consumption for Models Daily Over 14 Months

From the figure (4.6), a convergence was noticed between the curve of all models with each other and with the actual curve when implemented on test data. In spite of this, it is noticed that the LSTM is the closest to the actual consumption curve, while the curve of the Naïve Bayes model represented by green is the farthest from the real curve and the curves of other models.

4.5.2 Short Term Prediction (one week)

` To show the short-term (one week) predictions, In table (4.10) The results are shown. In comparison to KNN, SVR and Naïve Bayes, LSTM has shown to perform better for mid-term predictions. However, SVR have also shown close performances to LSTM, although it had the most training time.

Table (4.10) Short-Term Predictions (One Week)

model	Error			Time (second)
	MAE	MAPE	RMSE	
LSTM	0.145	15.182	0.179	11.61310
KNN	0.170	16.043	0.213	0.12974
SVR	0.147	14.870	0.179	0.15064
Naïve Bayes	0.199	19.018	0.254	0.09456

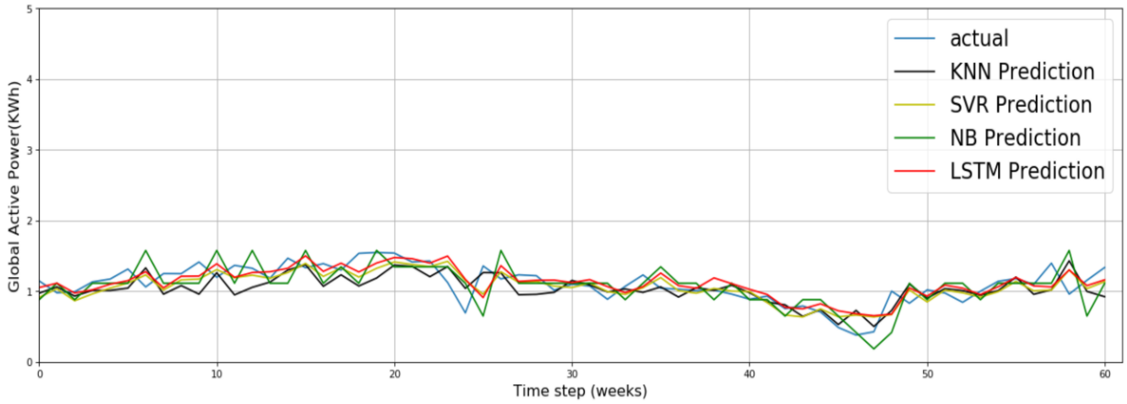


Figure (4.7) Actual Consumption vs. Predicted Consumption for Models Weekly Over 14 Months

Figure (4.7) shows the evolution of the forecast of the different models compared to the actual consumption for short term (one week), for which the LSTM faithfully follows the actual curve.

4.5.3 mid-term Prediction (one month)

To show the mid-term prediction, table (4.11) shows the mid-term power consumption. the LSTM model have outperformed the rest of the models for mid-term predictions although it had the most training time. However, SVR has also shown close performances to LSTM, with less training time compared to other models.

Table (4.11) Mid-Term Predictions (One Month)

model	Error			Time (second)
	MAE	MAPE	RMSE	
LSTM	0.145	14.018	0.166	5.833945
KNN	0.171	15.750	0.198	0.000967
SVR	0.160	15.301	0.176	0.072627
Naïve Bayes	0.171	17.413	0.216	0.001388

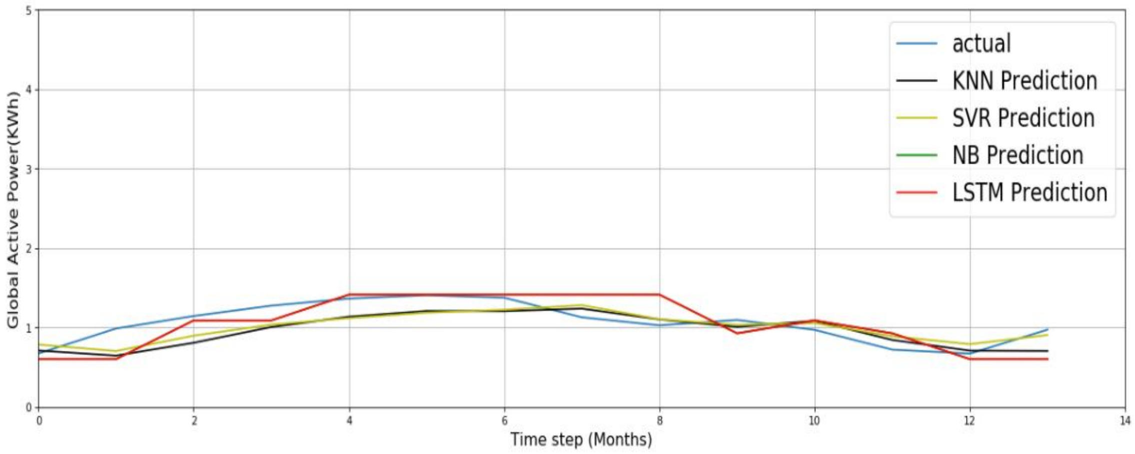


Figure (4.8) Actual Consumption vs. Predicted Consumption for Models Monthly Over 14 Months

Figure (4.8) shows the results of the mid-term (one month) prediction obtained by the four models (LSTM, KNN, SVR, Naïve Byes). From the extracted curves it is clearly noticed that the LSTM is the closest to the actual consumption curve. It is followed by the SVR. Then, slightly less so the KNN curves, while the Naïve Byes is the furthest from the real curve.

Although the implementation time for LSTM system is long compared to the rest of the systems, it is clear that LSTM system is capable of predicting the consumption of electricity in the short term (one day, one week), medium term (one month) with high accuracy.

4.6 Proposed Model vs. Related Work

By comparing the performance results of the proposed methodology against what is related to the work that has been proposed in the same field, it was found that the work of Kim and Cho [14] predicts electricity consumption and uses the same data set used by the proposed system, but applies a pre-treatment to it in a 60-minute window by the sliding window algorithm. It uses the CNN-LSTM algorithm to predict electricity consumption at time intervals per minute, per hour, per day, and per week, while the proposed system uses min - max normalization in pre-processing and LSTM algorithm on time ranges (one day, one week) short term and (one month) mid-term. Using the measures of accuracy: root mean square error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE) and comparing results, it is found that the proposed system may outperform existent work in the time period (one day, one week) and according to the Table (4.12):

Table (4.12) Comparison between Other Existing Work and The Proposed Work.

model	period	Error		
		MAE	MAPE	RMSE
Proposed Model (LSTM)	Daily	0.183	18.324	0.244
	Weekly	0.145	15.182	0.179
CNN-LSTM	Daily	0.257	31.83	0.322
	Weekly	0.238	31.84	0.309

Chapter Five

Conclusions and Suggestions for Future Works

Chapter Five

Conclusions and Suggestions for Future Works

5.1 Conclusions

Smart metering technologies enhance data analytics in energy management, and create new capacities for energy services. Predicting energy consumption in homes is an aspect of energy management, accounting for consumption in the residential sector, a large proportion of the total demand for electricity.

The study focuses on models that can be used to predict electricity consumption for an individual home for the short term (one day, one week) and the medium term (one month). The study compares the LSTM, KNN, SVR, Naïve Bays by conducting experiments with data for an individual house smart meter dataset in France. although the SVR showed good performance for short term predictions (one week). The LSTM showed similar performance to the SVR performance of short-term predictions, while it outperformed all other models in short-term predictions (one day, one week) and medium-term predictions (one month), while the naive performed the weakest compared to other models. It is evident that the LSTM is able to predict short-term (one day, one week) and medium term (one month) of electricity consumption with high accuracy.

5.2 Suggestions for Future Works

The future work would involve:

- The proposed model can be applied on a different dataset with varied input such as temperatures and environmental factors to enhance model performance.
- Utilization of model output to predict the cost of power and to determine the suitable times for maintenance by power suppliers.
- Other source of energy can be taken into consideration such as solar energy or biogas as source of energy to be predicted by the model effectively.
- Use of hybrid algorithms with the model to attempt obtaining more accurate results.

References

References

[1] Owusu, P. A., & Asumadu-Sarkodie, S. “A review of renewable energy sources, sustainability issues and climate change mitigation”. Cogent Engineering, 3(1), 1167990, (2016).

[2] Foucquier, A., Robert, S., Suard, F., Stéphan, L., & Jay, A. (2013). “State of the art in building modelling and energy performances prediction: A review”. Renewable and Sustainable Energy Reviews, 23, 272-288 (2013).

[3] Deb, C., Zhang, F., Yang, J., Lee, S. E., & Shah, K. W. “A review on time series forecasting techniques for building energy consumption”. Renewable and Sustainable Energy Reviews, 74, 902-924, (2017).

[4] Ghasempour, A. “Internet of things in smart grid: Architecture, applications, services, key technologies, and challenges”. Inventions, 4(1), 22, (2019).

[5] Bianco, V., Manca, O., & Nardini, S. “Electricity consumption forecasting in Italy using linear regression models”. Energy, 34(9), 1413-1421, (2009).

[6] Boukoros, S., Nugaliyadde, A., Marnerides, A., Vassilakis, C., Koutsakis, P., & Wong, K. W. “Modeling server workloads for campus email traffic using recurrent neural networks”. In International Conference on Neural Information Processing (pp. 57-66). Springer, Cham, (2017, November).

[7] Wang, Y., Chen, Q., Hong, T., & Kang, C. “Review of smart meter data analytics: Applications, methodologies, and challenges”. IEEE Transactions on Smart Grid, 10(3), 3125-3148 (2018).

[8] Zheng, J., Xu, C., Zhang, Z., & Li, X. “Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network”. In 2017 51st Annual Conference on Information Sciences and Systems (CISS) (pp. 1-6). IEEE. (2017, March).

[9] Quek, Y. T., Woo, W. L., & Logenthiran, T. “A naïve Bayes Classification Approach for Short-Term Forecast of Photovoltaic System”. Proceedings of the Sustainable Energy and Environmental Sciences, Singapore, 6-7. (2017).

[10] Fayaz, M., & Kim, D. “A prediction methodology of energy consumption based on deep extreme learning machine and comparative analysis in residential buildings”. Electronics, 7(10), 222, (2018).

[11] Gökgöz, F., & Filiz, F. “Deep Learning for Renewable Power Forecasting: An Approach Using LSTM Neural Networks”. International Journal of Energy and Power Engineering, 12(6), 416-420 (2018).

[12] Zhang, X. M., Grolinger, K., Capretz, M. A., & Seewald, L. “Forecasting residential energy consumption: Single household perspective”. In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 110-117). IEEE. (2018, December).

[13] George, S., Almond, V., James, E., Vishwakarma, S., & Kiruthika, M. "Analysis of Electric Power Consumption using Smart Meter Data", (2018).

[14] Kim, T. Y., & Cho, S. B. “Predicting residential energy consumption using CNN-LSTM neural networks”. Energy, 182, 72-81. (2019).

[15] Adewuyi, S. A., Aina, S., & Oluwaranti, A. I. “A deep learning model for electricity demand forecasting based on a tropical data”. Applied Computer Science, 16(1). (2020).

[16] Solyali, D. “A Comparative Analysis of Machine Learning Approaches for Short-/Long-Term Electricity Load Forecasting in Cyprus”. Sustainability, 12(9), 3612, (2020).

[17] Bimenyimana, S., & Asemota, G. “Traditional Vs Smart Electricity Metering Systems: A Brief Overview”. Journal of Marketing and Consumer Research, 46, (2018).

[18] Metering & Smart Energy International (2006). The history of the electricity meter. Available at: https://www.metering.com/features/the-history-of-the-electricity-meter/ (Accessed 2020-2-20).

[19]https://www.electricmetersales.co.uk/product/iskra-e89e2-single-phase/(Accessed: 2020-02-08).

[20] International Energy Commission (IEA). How2Guide for Smart Grids in Distribution Networks, Roadmap Development and Implementation. 2016. Available online: https://www.iea.org/reports/how2guide-for-smart-grids-in-distribution-networks(Accessed: 2020-2-20).

[21] Hossain, M. R., Oo, A. M. T., & Ali, A. S. "Evolution of smart grid and some pertinent issues". In 2010 20th Australasian Universities Power Engineering Conference (pp. 1-6). IEEE (2010, December).

[22] Saleem, Y., Crespi, N., Rehmani, M. H., & Copeland, R. "Internet of things-aided smart grid: technologies, architectures, applications, prototypes, and future research directions". IEEE Access, 7, 62962-63003(2019).

[23] Weranga, K. S. K., Kumarawadu, S., & Chandima, D. P. "Smart metering design and applications". Singapore: Springer (2014).

[24]https://SiteCollectionDocuments/SmartEnergy/HowToReadASmartMeter.pdf/(Accessed: 2020-02-08).

[25] AlAbdulkarim, L. O., & Lukszo, Z. "Smart metering for the future energy systems in the Netherlands". In 2009 Fourth International Conference on Critical Infrastructures (pp. 1-7). IEEE (2009, March).

[26] Deb, S., Bhowmik, P. K., & Paul, A. "Remote detection of illegal electricity usage employing smart energy meter-A current based technique". In ISGT2011-India (pp. 391-395). IEEE (2011, December).

[27] Toledo, F. "Smart metering handbook". PennWell Books, (2013).

[28] Sun, H., Poor, H. V., Hatziargyriou, N. D., & Carpanini, L. (2016). “Smarter energy: from smart metering to the smart grid”. Institution of Engineering & Technology (2016).

[29]http://www.Smartgrid.gov/the_Smart_grid#Smart_home (Accessed: 2020-02-09)

[30] Al-Waisi, Z., & Agyeman, M. O. “On the challenges and opportunities of smart meters in smart homes and smart grids”. In Proceedings of the 2nd International Symposium on Computer Science and Intelligent Control (pp. 1-6), (2018, September).

[31] Ghasempour, A. “Optimum packet service and arrival rates in advanced metering infrastructure architecture of smart grid”. In 2016 IEEE Green Technologies Conference (GreenTech) (pp. 1-5). IEEE (2016, April).

[32] Ghasempour, A., & Moon, T. K. “Optimizing the number of collectors in machine-to-machine advanced metering infrastructure architecture for internet of things-based smart grid”. In 2016 IEEE Green Technologies Conference (GreenTech) (pp. 51-55). IEEE (2016, April).

[33] Al-Hamadi, H. M., & Soliman, S. A. “Long-term/mid-term electric load forecasting based on short-term correlation and annual growth”. Electric power systems research, 74(3), 353-361, (2005).

[34] Han, J. "M. Kamber och J. Pei, “Data Mining: Concepts and Techniques”, Waltham. (2012).

[35] Rahman, A., Srikumar, V., & Smith, A. D. “Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks”. Applied energy, 212, 372-385, (2018).

[36] Mahmud, B. N., Ferdoush, Z., & Mim, L. T. (2019). Modelling and Forecasting Energy Demand of Bangladesh using AI based Algorithms (Doctoral dissertation, Brac University).

[37] Li, T., Wang, B., Zhang, L., & Zhao, X. “Short-term load forecasting using optimized LSTM networks based on EMD. In 2018 10th International Conference on Communications, Circuits and Systems (ICCCAS) (pp. 84-88). IEEE (2018, December).

[38] Zurada, J. M. “Introduction to artificial neural systems” (Vol. 8). St. Paul: West. (1992).

[39] Kingma, D. P., & Ba, J. “Adam: A method for stochastic optimization”. arXiv preprint arXiv:1412.6980, (2014).

[40] Ruder, S. “An overview of gradient descent optimization algorithms". arXiv preprint arXiv:1609.04747, (2016).

[41] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. “Improving neural networks by preventing co-adaptation of feature detectors”. arXiv preprint arXiv:1207.0580. (2012).

[42] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. “Dropout: a simple way to prevent neural networks from overfitting”. The journal of machine learning research, 15(1), (2014).

[43] Vijayakumar, S., & Schaal, S. “Local dimensionality reduction for locally weighted learning”. In Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation' (pp. 220-225). IEEE (1997, July).

[44] Martínez, F., Frías, M. P., Charte, F., & Rivera, A. J. “Time Series Forecasting with KNN in R: the tsfknn Package”. The R Journal, 11(2), 229-242, (2019).

[45] Shoesmith, E. “Estimation of Dependences Based on Empirical Data”. Journal of the Royal Statistical Society: Series D (The Statistician), 33(3), 324-324, (1984).

[46] Jain, R. K., Smith, K. M., Culligan, P. J., & Taylor, J. E. “Forecasting energy consumption of multi-family residential buildings using support vector regression: Investigating the impact of temporal and spatial monitoring granularity on performance accuracy”. Applied Energy, 123, 168-178 (2014).

[47] Ma, Z., Ye, C., & Ma, W. “Support vector regression for predicting building energy consumption in southern China”. Energy Procedia, 158, 3433-3438 (2019).

[48] Parrella, F. “Online support vector regression”. Master's Thesis, Department of Information Science, University of Genoa, Italy, 69 (2007).

[49] Brownlee, J. “Master Machine Learning Algorithms: discover how they work and implement them from scratch”. Machine Learning Mastery (2016).

[50] Garg, B. “Design and Development of Naive Bayes Classifier” (Master of Science Thesis). North Dakota State University, North Dakota (2013).

[51] Kassa, Y.; Zhang, J.; Zheng, D.;Wei, D. “Short term wind power prediction using ANFIS. In Proceedings of the 2016 IEEE International Conference on Power and Renewable Energy (ICPRE)”, Shanghai, China, 21–23 (October 2016).

[52]https://archive.ics.uci.edu/ml/datasets/individual+household+electric+power+Consumption.

Appendix A

1. Global active power: The active power that the household consumes per minute (in kilowatt).

2. Global reactive power: The reactive power that the household consumes per minute (in kilowatt).

3. Voltage: averaged voltage per minute (in volt).

4. Global intensity: per minute, the averaged current intensity (in ampere).

5. Sub metering 1: energy (in watt-hour of active energy). It corresponds to the kitchen, containing mainly a microwave, an oven and a dishwasher (here hot plates are gas powered not electric).

6. Sub metering 2: energy (in watt-hour of active energy). It corresponds to the laundry room, containing a tumble-drier, a washing-machine, a light and a refrigerator.

7. Sub metering 3: active energy (in watt-hour of active energy). for an air-conditioner and an electric water-heater.

Sample of the dataset [57]

Date	Time	Global active power	Global Reactive power	Voltage	Global intensity	Sub Metering 1	Sub Metering 2	Sub Metering 3
16/12/2006	17:24:00	4.216	0.418	234.840	18.400	0.000	1.000	17.000
16/12/2006	17:25:00	5.360	0.436	233.630	23.000	0.000	1.000	16.000
16/12/2006	17:26:00	5.374	0.498	233.290	23.000	0.000	2.000	17.000
16/12/2006	17:27:00	5.388	0.502	233.740	23.000	0.000	1.000	17.000
16/12/2006	17:28:00	3.666	0.528	235.680	15.800	0.000	1.000	17.000

الخلاصـة

يؤدي النمو السكاني المتزايد وعدد الأجهزة الكهربائية المستخدمة يومًا بعد يوم إلى زيادة استهلاك الطاقة الكهربائية ، وبالتالي الطلب على الكهرباء ، مما يؤدي إلى الضغط على موردي الكهرباء. نظرًا لوجود العديد من العوامل التي تؤثر على استهلاك الكهرباء ، فإن استخدام تقنية العداد الذكي يمكّننا من الحصول على كميات هائلة من البيانات على مدار الساعة ، وهذا يسهل التنبؤ باستهلاك الطاقة وتنظيم إدارة الطاقة.

في هذه الدراسة ، تم اقتراح نموذج للتنبؤ باستهلاك الطاقة لمنزل واحد على المدى القصير (يوم واحد ، أسبوع واحد) والمدى المتوسط (شهر واحد). يتكون النموذج المقترح من أربع مراحل: مرحلة قراءة البيانات ومرحلة ما قبل المعالجة ومرحلة التنبؤ ومرحلة تقييم الأداء. تم جمع مجموعة البيانات من خلال عداد ذكي من منزل واحد للتحقق من صحة النموذج وتحليل النتائج. ثم ، تم تطبيق آلة التعلم العميق (Long Memory-Term Memory LSTM) وخوارزميات التعلم الآلي المعروفة وهي خوارزميه(Vector Regression SVR)، و خوارزميه(K-nearest neighbor KNN) وخوارزمية(Naive Bayes) على البيانات المعالجة مسبقًا للتنبؤ باستهلاك الطاقة ليوم واحد ، أسبوع واحد وشهر واحد. وقد تمت مقارنتها باستخدام المقاييس الإحصائية: متوسط الخطأ المطلق (MAE) ، متوسط الخطأ النسبي المطلق (MAPE) وجذر متوسط الخطأ التربيعي (RMSE) لقياس أداء خوارزميات التعلم الآلي هذه.

تشير قيم القياس الإحصائية هذه إلى أن أداء النموذج المقترح LSTM أفضل من K-NN و SVR و Naïve Bayes للتنبؤ بالطاقة ليوم واحد وأسبوع وشهر واحد على البيانات المقدمة. وكانت نتائج نموذج LSTM هي MAE 0.183 و MAPE 18.324 و RMSE 0.244 للتنبؤ باستهلاك الطاقة ليوم واحد و MAE 0.145 و MAPE 15.182 و RMSE 0.179 للتنبؤ باستهلاك الطاقة لمدة أسبوع واحد. MAE 0.145 و MAPE 14.018 و RMSE 0.166 للتنبؤ باستهلاك الطاقة لمدة شهر واحد. من الواضح أن نموذج LSTM قادر على التنبؤ باستهلاك الكهرباء على المدى القصير (يوم واحد ، أسبوع واحد) والمدى المتوسط (شهر واحد) بدقة عالية.

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة ديالى
كلية العلوم

تحليل مجموعة بيانات استهلاك الطاقة الكهربائية بناءً على نماذج التوقع

رسالة مقدمة
الى كلية العلوم في جامعة ديالى وهي جزء من متطلبات نيل شهادة الماجستير في علوم الحاسبات

تقدمت بها الطالبة

زهراء جبار زغير

بإشراف

أ.د. ظاهر عبد الهادي عبد الله

1441 هـ   2020 م