Modelling and Forecasting Energy Demand of Bangladesh using AI based Algorithms

by

Booshra Nazifa Mahmud
15301020
Zannatul Ferdoush
15301068
Lamia Tasnim Mim
15301052

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
BRAC University
April 2019

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at BRAC University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Booshra Nazifa Mahmud
15301020

Zannatul Ferdoush
15301068

Lamia Tasnim Mim
15301052

Approval

The thesis/project titled “Modelling and Forecasting Energy Demand of Bangladesh using AI based Algorithms” submitted by

1. Booshra Nazifa Mahmud (ID: 15301020)
2. Zannatul Ferdoush (ID: 15301068)
3. Lamia Tasnim Mim (ID: 15301052)

Of Spring, 2019 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on April 25, 2019.

Examining Committee:

Supervisor:
(Member)

Dr. Amitabha Chakrabarty
Associate Professor
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Dr. Jia Uddin
Associate Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Dr. Md. Abdul Mottalib
Professor
Department of Computer Science and Engineering
BRAC University

Abstract

Bangladesh being one of the five fastest growing economies in the world with its enormous 164.7 million population is facing a huge challenge of adapting to the surging demand of electricity rising throughout the country to support its blooming economy. Load forecasting can play a vital role to overcome this challenge as it serves as an imperative tool behind electric utilities planning and operation management. Forecast loads serve as the basis of many operational decisions such ensuring maximum utilization of power by avoiding under or over generation; understanding future load demand to make economically viable investment decisions; management of resources; infrastructure development along with maintenance schedule planning. This study, first of all, proposes an automated model that fetches data from daily load generation reports (kept in pdf format) found in the Bangladesh Power Development Boards website [54] to generate a compact dataset of our historical load data with which we have conducted this research. The necessity of this model is no publicly available dataset have been found so far that contains the historical load data along with data of other important features that effect the forecast load. Secondly, we have approached three major machine learning methods – K Nearest Neighbor, Random Forest, and Long Short Term Memory (LSTM) to observe how these algorithms perform in forecasting load based on the historical electric load data of Bangladesh and what features play important roles to accurately forecast load. We have found that among these three algorithms; LSTM yields the best result having minimal prediction error compared to the other algorithms.

Keywords: Load Forecast; Machine Learning; Prediction; Decision tree; Random Forest; K Nearest Neighbour; Long Short Term Memory

Acknowledgement

We would like to take this opportunity to acknowledge and express our sincere gratitude and appreciation to the people who have given us their assistance throughout the preparation of this research. We would especially like to thank our thesis supervisor, Dr. Amitabha Chakrabarty for his continuous encouragement as well as creative guidance in this field of research.
We are also grateful to our parents for being our constant support and inspiration through out this whole journey.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AI Artificial Intelligence

ANN Artificial Neural Network

AR Autoregressive

$ARIMA$ Autoregressive Integrated Moving Average

$BPDB$ Bangladesh Power Development Board

ES Exponential Smoothing

KNN K Nearest Neighbor

$LEAP$ Long-range Energy Alternative Planning

$LSTM$ Long Short Term Memory

MA Moving Average

MSE Mean Squared Error

$RMSE$ Root Mean Squared Error

RNN Recurrent Neural Network

$SLTF$ Short Term Load Forecasting

SVM Support Vector Machine

SVR Support Vector Regression

Chapter 1

Introduction

To measure power system performance, load forecasting has always been considered as one of the key elements. The term 'Load Forecasting' refers to the procedure of predicting the electrical power required to meet the short or long term demand of customers; a strategy that is utilized by energy utilities to foresee the amount of power or energy needed to fulfill the existing need and supply balance among the consumers [43].

The ultimate goal of any power generation company is to distribute quality energy to its customers with proper security while upholding the economic values of the company. To ensure this, most of the electric company face many challenges in terms of operating, managing and controlling the electric energy system [8]. Fan and Chen [22] discussed that load forecasts serve as the basis to overcome these challenges through giving guidance in taking operation decisions such as dispatch scheduling of generating capacity, reliability analysis, and maintenance plan for the generators.

In competitive electricity markets, energy transactions are crucially dependent on forecast loads. The forecast of electricity price is also dependent on accurately forecast load [22]. According to Hahn et al. [29], the necessity of reliable load forecast is more important than ever as most of the energy markets around the world are becoming deregulated. Particularly in the present time, the presence of deregulated electric power industry with the support of free competitive market has made load forecasting more crucial than ever. To make a stand in the deregulated market competition, the first thing any participant needs is the solid idea of how much electricity will be needed over a particular period of time in the future. Hahn et al. [29] described that the operational cost may rise remarkably as a outcome of meeting excess demand of electricity within a short time. It happens when the supplier underestimate the energy demand. On the other hand, the occurrence of resource waste is seen in the case where the supplier overemphasize the energy demand [18].

Furthermore, the price of electricity is heavily dependent on the forecast load demand. As a result, load forecasting serves as the decisive factor behind most of the major conclusions made in the electricity market. The accuracy of forecasting is of incredible importance for the planning and operations of organizations like financial institutions, regional transmission organizations, energy suppliers, and other participants responsible in the generation, transmission, and distribution of electricity. To add to that, market share and profit along with shareholder value are gravely manip-

ulated by the forecast error [22]. Due to the immense importance load forecasting holds in controlling the electricity market, many researches have been conducted on this field since very long time. Gross and Galiana [5] in their survey paper published in 1987 discussed about a handsome number of publications on load forecasting and analysis; some dating back as far as 1966 .
Given the above mentioned scenarios, we can state that electrical load estimation is an imperative procedure that can increase the effectiveness and incomes of the companies that produce and distribute electricity. The forecasting also helps the utility companies in their operation and management of the supply of electricity to their customers. It helps them to anticipate their ability and activities so as to dependably supply all purchasers with the required vitality [43].

1.1 Energy Revolution in Bangladesh

The rate of energy consumption of any society directly refers to its magnitude of development. The more developed a country is, the more its demand for energy grows. Therefore it is very evident that, energy is also a crucial input parameter to evaluate economic development of a country. Bangladesh, a very young country having gained its independence in the year 1971, has seen tremendous change in its revolution towards energy consumption in the the past decades since its independence.
The power sector has experienced considerable progress in meeting the demand for electricity. As described in [56], access, coverage, and level of consumption have significantly increased over the years. Daily load shedding has significantly dropped: from 1107 MKWH in 2009 to 32 MKWH in 2018. However, Bangladesh's coverage and access are still behind regional standards (Bangladesh vs. South Asia average: 76 per cent vs. 85.6 per cent in 2016). Besides, energy use efficiency has been improving. According to the [42], energy use per GDP (kg OE/1000 US $) has reduced from 307 kg OE in 2007 to 218 kg OE in 2014. This is happened owing to strong economic growth backed by the 5 expansion of less-energy-intensive export industries, such as RMG sector.
The power sector has been able to come out from the period of crisis [42]. The equity aspects of demand for power is still a major concern. Government has undertaken 'Upazilla 100% Electrification Program' under which 256 upazillas have been electrified by September, 2018. Another 63 upazillas were targeted to be electrified by December, 2018 and the rest 142 upzillas to be covered by December, 2019. Poor quality of electricity supply affects the economic activities of different parts of the country. According to the newspaper report, frequent outages of electricity outside major cities severely affect economic activities in districts such as Rangpur, Barisal, Bhola, Rajshahi, Noakhali, Narail, Natore, Bagerhaat, Naogan, Rajbari, Sylhet and Bogra. The quality of electricity supply is measured by standard power factor. Due to poor power factor, the distribution companies were penalized. For example, the Dhaka Power Distribution Company Limited (DPDC) were faced power factor charges (PFC) (fine) by about Tk.841 crore during 2013-17 due to failure to maintain the standard.
According to the anecdotal information, foreign investors who invested in electricity-intensive manufacturing industries expressed their concerns about poor quality of electricity. The future demand for electricity in the country will be changed. Accord-

ing to the Power Sector Master plan (PSMP) [42], the structure and composition of electricity demand will be changed with the changes in economic activities.
There is significant rise in demand in industry and commercial and public services (e.g. special economic zones; metro rails and other services). The projected peak electricity demand in coming years (base case) would be 14,500 MW (in 2021), 27,400 MW (in 2030), and 51,000 MW (in 2041) respectively [42]. According to the Power Sector 7 Efficiency Master Plan, government has set the target to improve energy intensity by 20 per cent by 2030 compared to the 2013 level. Overall, addressing the future demand will need a shift in the demand management. More focus is needed to put on quality of electricity and is required to put emphasis on improvement of users' efficiency.
Present reservoirs (e.g., natural gas, coal, and hydro power) of primary commercial energy sources in Bangladesh produce very modest amount of energy compared to the prevailing demand in the country [30]. Our main source behind power generation is fossil fuel (natural gas) which measured to 81.4% in producing total installed electricity generation capacity (5248MW) in 2006 [30]. Only about 42% people of our country had access to electricity on that year while the majority suffered from power deprivation [28].
The current working government of Bangladesh has declared their aim to provide electricity for all by the year 2020 as one of the main target to achieve "Digital Bangladesh". Although at present there is still highly unsatisfied demand for energy [28]. Since Bangladesh has a expeditiously growing economy, the electricity demand is also growing rapidly. The share of electricity in the final stage of consumption has also been increasing due to system loss, lateness to complete new power plants, making low efficient of power stations, electricity theft, collapse and ill maintenance of the power plants. To overcome these drawbacks, load forecasting can prove useful by predicting final electricity demand considering the drawbacks to lessen the share of electricity in final stage of consumption.

1.2 Categories of Load Forecasting

Load forecasts can be divided into three categories: short-term forecasts which usually forecast from one hour to one week, medium forecasts that range from a week to a year, and long-term forecasts which are longer than a year. For this research, we have used short-term load forecasting (STLF). Short-term forecasts have risen to become important since the energy markets around the world started becoming free and competitive [8]. Like many countries around the world, Bangladesh has also partially privatized its power systems, and as a result of the deregulated market - electricity has now become an object to be sold and bought at market prices. Due to the significant role of forecast load in determining the market prices, load forecasting has become discernibly important to the power industry.
Load forecasting is however a difficult task and especially in short-term. Short-term load forecasting serves as the base for safe and economic operation of power systems. Influential variables like time of the year, weather and price of electricity etc and their relationship with the load has a vital role in predicting the future short term load. We can choose multiple target values as forecasting goal in short term load forecasting. Some system forecasts peak loads while other forecasts cumulative loads of a certain period of time.

Load forecast is also done with different features of the electricity network. The network managers as well as utility companies must know how much energy has to be generated the next day and other important factors such as maximum demand, total fuel needed to operate the plants to plan efficiently. This is where short term forecasting comes handy by forecasting electric load 24 hours to a week ahead by providing an understanding of the demand. As it deals with the demand of immediate next day or week, the forecasting needs to deliver result with minimum error so no shortage or wastage will occur. Our challenge is to figure out the features that have maximum effect on our load data and to compare the results of forecast load data of different models to observe the most promising outcome with the machine learning approach.

1.3 Advantages of Load Forecasting

1. The electricity utilities get the advantage to design power plants skillfully since they have a comprehension of the future consumption or load demand.

2. Understanding the future load in long haul encourages the concerned organizations to plan and settle on financially suitable choices concerning future age and transmission speculations which minimize the risks for service organizations.

3. Load forecasting helps to identify areas or districts with high power consumption, and the utilities in all likelihood will create the power stations closer to the areas with maximum power consumption to minimize the transmission and distribution cost as well as the associated system losses.

4. Forecasting can assist to determine the required resources and plan for maintenance of the power system [43]. By understanding the demand with respect to time, the utility company can carry out the maintenance works accordingly and thus ensure minimum disturbance to the consumers.

1.4 Contribution Summary

In the past few decades, electric load forecasting has been widely experimented and scrutinized with different classical and more recently with varied artificial intelligence based techniques to reach towards such an effective system that can yield minimum error and maximum accuracy in terms of predicting future electricity demand. Even though these works have been conducted on the time series based historical data of different countries, there has been very little work found so far on the historical load data of Bangladesh to forecast load as well as in finding the internal as well as external factors having maximum effect on the load consumption using machine learning based algorithms.

In our research, we have tried to see how machine learning algorithms perform in predicting the load demand of our country in short-term (24 hours ahead) using the historical load data of Bangladesh. For this purpose, we have explored three popular techniques of machine learning that have been creating buzz in the recent times in the field of load forecasting using computational intelligence based techniques. The mentioned algorithms are - Random Forest, K- Nearest Neighbor and Long Short Term Memory. Along with that the significance of the features like temperature, seasonal data, amount of fuel (natural gas) used to produce the majority of the electricity have been analyzed to find the correlation and their impact factor on load consumption. Finally, we have tried to examine if machine learning approaches aid in finding hidden patterns in our dataset and how these patterns effect the forecast result.

1.5 Thesis Outline

In this paper, Chapter 2 provides the literature review in details including the varied algorithms and techniques that has been explored to forecast energy demand in various researches and experiments. Chapter 3 presents the methodologies that have been used in this research to predict load demand, and a detailed analysis of data and important factors for forecast as well as experimental setup including the algorithm implementation and techniques are discussed in Chapter 4 alongside the results and analysis. Lastly, Chapter 5 gives the conclusion and future work.

Chapter 2

Literature Review

Research for short time load forecasting goes way back. As a result we can see various methods applied in order to solve this issue. Statistical analysis, methods of time series analysis [37], Box Jenkins [4], regression analysis and many other methods have been used in order to produce a noteworthy forecasting model. In recent times, Artificial Intelligence (AI) based methods such as fuzzy logic system, expert systems and Artificial Neural Network (ANN) are the most widely used approaches around the world for load forecasting. Neural network models in particular, have been used in different hybrid approaches to develop an efficient load forecasting model. Since we developed and improved our investigation methods using appropriate mathematical tools it has led to more accurate load forecasting techniques.
Cartesian Genetic programming was used to develop a Recurrent Neural Network (RNN) model [37]. The major intention of this work was to predict the peak loads for the next day in advance using the historical load data of the last 10 days. Another method named Long-range Energy Alternative Planning (LEAP) tool was used in the paper by Mondal et al. [30]. They worked in order to calculate the electricity demand for different sectors up to the year 2035 considering the base year 2005. LEAP is a defined as a computerized framework for the assessment of regional energy planning strategies. The LEAP methodology can be classified as a bottom-up approach and it interrelates electricity demand with GDP, urbanization, technical change, etc. Again the paper by, [26] presents three modeling techniques for the prediction of electricity energy consumption. Decision tree and neural networks are also considered along with traditional regression analysis. The accuracy in predicting electricity energy consumption of Hong Kong is compared in this analysis. They compared three different techniques: regression analysis, decision trees, and neural networks.
In this chapter, the next section reviews statistical and regression approaches including basic time series approach. After that the most discussed and popular Machine Learning approaches will be described so that we can justify why we moved towards the modern computation methods.

2.1 Classical Approaches

Generally statistical approaches require a mathematical model that represents load as a function of time, weather, or customer class. Additive models and multiplicative models are two important categories of such mathematical models. They differ in the type of the forecast load. For example, the following function of four components:

$$L = L_n + L_w + L_s + L_r \tag{2.1}$$

here L is the total load, L_n represents the "normal" part of the load, which is a set of standardized load shapes for each "type" of day that has been identified as occurring throughout the year, L_w represents the weather sensitive part of the load, L_s is a special event component that create a substantial deviation from the usual load pattern, and L_r is a completely random term, the noise. A multiplicative model may be of the form

$$L = L_n * F_w * F_s * F_r \tag{2.2}$$

where L_n is the normal (base) load and the correction factors F_w, F_s, and Fr are positive numbers that can increase or decrease the overall load. These corrections are based on current weather (F_w), special events (F_s), and random fluctuation (F_r). Factors such as electricity pricing (F_p) and load growth (F_g) can also be included. In the paper by Rahman [6] a rule based forecast using a multiplicative model was presented. Weather variables and the base load associated with the weather measures were included in the model. Another way to classify the statistical approach is stationary and non-stationary techniques. Non-stationary methods used for forecasting electric loads that exhibit a trending pattern, includes the regression method.

Regression based approach is the one of the earliest and most widely used statistical techniques. According to Soliman and Al-kandari [32] the dominant variables are identified on the basis of correlation analysis with load. It was also pointed that the significance of those variables are determined through statistical tests. Mathematically, the load model using this approach can be written as

$$y(t) = a_0 + \sum_{i=1}^{n} (a_i x_i(t) + r(t)) \tag{2.3}$$

where y(t) is the load value at time t, x_1(t), . . . , x_n (t) are explanatory variables, r(t) is the residual load at time t, and a_i are the regression parameters relating the load y(t) to the explanatory variables. Previous analysis that uses this model treats a_i as a crisp number.

A study by Al-Fuhaid [13] points out that, the time series analysis is not a weather sensitive approach rather it uses historical load data for speculation of future load. It is also mentioned that in the stochastic time-series method, the load is modeled as the output of a linear filter driven by white noise, [32]. The autoregressive (AR) and moving average (MA) processes are the two very simple forms of stochastic analysis of time series data. Both of these methods use the time and load as the only input parameters. The lack of weather input into time-series models typically limits their forecasting ability.

The Holt-Winters exponential smoothing (ES) [36] and Autoregressive Integrated Moving Average (ARIMA) models [27] are most commonly employed conventional STLFs. The time series is broken down into two elements: trend and seasonal components, in exponential smoothing. ARIMA models can be extended for the case of multiple seasonalities but selecting appropriate model orders is an inconvenience. This order selection process is considered difficult to apply and also subjective.

2.2 Machine Learning Based Approaches

In all these years of researches to forecast energy demand when statistical approaches were not able to forecast load with better results as the load data is highly non-linear, modern approaches were suggested. Many machine learning or artificial intelligence techniques have been applied to predict electricity demand in Short Term Load Forecasting. These are mainly Regression Algorithms, Fuzzy Logic systems, and Neural Networks. In the next sections we will discuss the previous work done on electricity demand forecasting using machine learning approaches.

2.2.1 Regression Approaches

Regression Algorithms have been used to orchestrate short-term electricity-load forecasting in many experiments and studies over the past few decades. Many of this regression methods are used for both classification and regression analysis. The main challenge is to feed real world time series data into these models and train them with all the necessary features of data set. While many regression techniques perform surprisingly well, many disappoint to achieve expected result.

Random Forest

In the study by Huang et al. [44] proposes a permutation importance-based feature-selection method to conduct short-term electricity-load forecasting using the random forest algorithm. The original feature set was used to train the random forest model. Random forests have been used for STLF in another study [35] but the novelty of this work is data preprocessing. In a article [39] the random forest was studied as a univariate model for where they use real world load data to provide an example of model building and forecasting in practice.

Support Vector Machine

Machines learning algorithms, such as support vector machines (SVMs) have been observed as a reassuring alternative, for both regression [9] and classification [15]. A tutorial about SVMs application in pattern classification problems has been introduced [14] . Extending this technique to deal with regression problems, the SVM approach has been considered highly competitive, and it is also possible to highlight the applications involving time-series forecasting [11]. However, it is observed that the SVM felicitousness has been impeded by the essentiality of choosing the (i) the correct kernel function ; (ii) optimal parameters ; (iii) the loss function and (iv) trade-off parameter C [24]. Their work was to apply SVMs, varying the kinds of kernels, to predict the load of the next 24h (MW/h) meaning of the next day. then again a support-vector regression model was proposed to calculate the demand-response baseline for short-term electrical-load forecasting of office buildings[51]. Support Vector Regression (SVR) was used for electrical load forecasting, in a competition organized by EUNITE network (European Network on Intelligent Technologies for Smart Adaptive Systems) [20].

K Nearest Neighbor

K Nearest Neighbor(KNN) is also known as a lazy learner. It was initially proposed as a simple but effective classification approach. KNN is quite effective for either classification or regression problems even though it has a very simple methodology. A KNN classifier is designed and has been used to assess stability of power systems [33]. The KNN algorithm is also applied for power quality disturbance classification [48]. In a study of the effects of the accuracy of hourly load forecasting on the generation planning and operation of an electric utility was presented. First, a KNN classification technique is proposed for load forecasting. Then, the hourly load forecasting errors are compared with those obtained results by using a M5' [19]. In a publication [47], an initial approach for such forecaster based on the K-nearest neighbours (KNN) method was proposed.

2.2.2 Fuzzy Logic System

Fuzzy logic methods are the alternative to the conventional methods of load forecasting. A fuzzy rule base is developed to produce 'fuzzy' forecasts and defuzzification is performed to generate a point estimate for system load. This methodology yields accurate results comparable to other more complex statistical models. In [49] reviewed the technology of load forecasting around the context of a fuzzy logic approach, and artificial neural network method combined. In [50] the temperature and the time is used as input to the fuzzy system. The hourly data of the temperature and the corresponding time are given through fuzzification block to the interface system. According to them flexible controllability and smooth operation of power system with high efficiency can be achieved by applying the proper fuzzy logic control. In [52] a method of hourly load forecasting using fuzzy logic has been presented. The load forecasting has been done using the one-year data from the large-scale power system. The aim of the work is to determine the probable load curve of a particular day observing one-year data from large-scale industry. According to their scenario the fuzzy logic model is working satisfactorily with permissible error. According to [34] Fuzzy logic also have a slow response due to serial data processing.

2.2.3 Artificial Neural Network

Artificial neural networks (ANN or simply NN) has been widely studied and used to predict electricity load. This technique is known as universal approximator. The main advantage of Neural networks is that the inputs could be the outputs of other network component. For this and many other reasons ANNs in short-term load forecasting has been recognized by researchers over the world. An introduction of adaptive pattern recognition and self-organizing techniques for short term load forecasting has gained a lot of attention[2]. Later, it was developed as an adaptive neural network for short term load forecasting[7]. Another important feature of ANN is that it is capable of learning the co-relation between past, current, and future variables. So, in the case with time series data, the ANN traces previous load patterns and predicts a load pattern using recent load data.

Real world load data and weather data from the Hydro-Quebec databases were used in this study as inputs to inspect ANN capabilities in short term load forecasting [21]. The inputs to the neural network were hour and day indicators, weather related inputs and historical load data. As load history of the last few hours is not available, estimated values of this load are used instead. However, one major limitation of this work was that a small error in these estimated values could grow dramatically and lead to a serious problem in load forecasting since the previous output is fed back to the network as an input in the load forecasting procedure. So, later ANN capabilities was demonstrated in load forecasting without the use of load history as an input [23]. In addition, only temperature was used here, because results showed that other weather variables such as sky condition and wind direction have no serious effect and not needed to be considered in the electricity load forecasting procedure.

Chapter 3

Methodology

This chapter highlights the systematic, theoretical analysis of the methods applied to our system developed to forecast electricity demand of Bangladesh. It consists the theoretical analysis of the methods and principles associated with a branch of knowledge. It also includes concepts such as theoretical model, phases and techniques. We avoided the application statistical methods as they were backdated and were unable to manipulate huge amount of data and understand the pattern of data. We moved towards analyzing machine learning algorithms as they are more trendy and provide more promising results. We studied both Regression algorithms and most popular Neural Network algorithms. We can see in some of the articles Artificial Neural Networks provided better results, some demand that Support Vector Machines provide a good result. Then again different papers approached different Neural Networking algorithms. But it was visible that it varies upon the pattern of data used in the experiment. We wanted to examine how our data responds to different machine learning algorithms.
After studying all the papers available on the field of study and analyzing their results we decided to implement our system by applying three algorithms. Firstly Random Forest was chosen as it pick random data and promise to identify data pattern more accurately. Secondly, we chose K Nearest Neighbors from Regression algorithm as it follows a very fundamental idea of predicting data. At last we chose Long Short Term Memory a type of Recurrent Neural Network as it follows the principle of brain cells for storing memory. In the next sections we will describe the analysis of the principles of methods of the mentioned algorithms.

3.1 Random Forest

The history of decision tree and its training procedure as well as how a collection of decision tree become Random Forest is discussed in the following sections.
Regression tree is used by many forecasters now a days to aptly forecast load demand [25]. In a decision tree, the training data is split into sub units of data as long as we cannot achieve acceptable result. We can see widespread usage of decision trees in many decision making automated sectors. Previously, rules that made up the decision tree were influenced by human command. But due to the time consuming nature of the process; the problems became more complex in nature. As a mean to solve this problem, automated rule extraction system was introduced. Hunt et al. [1] discusses one of the initial works on the field of automated decision tree creation.

As of today, many researches have been conducted to find out the factors to be considered in building a decision tree that performs with least error.
CART (Classification and Regression Trees) algorithm was proposed by Breiman et al. [3] in developing decision trees. As we know, load forecasting problem specify as a quantitative regression problem; therefore regressive decision trees are used to predict future loads. This is done by looking for the peak value of successive splits between the training dataset as long as the prediction error do not lessen to minimum scale.
As we can see in Lier [41], the root node of the decision tree stores all the training data. The process of decision tree building starts form this node. At this stage, we can see that the predicted value is equivalent to the average of all the target values found in the training data. The prediction error is calculated as Mean Squared Error (MSE) using the following equation where N_l is the number of training samples in l, D_l iterates over all samples in l, y_i is the target value at sample i, and y_l is the predicted value for this leaf node.

$$MSE(l) = 1/N_l \sum_{D_l} (y_l - y_i)^2 \tag{3.1}$$

If we want to lessen prediction error, the split in the training set must be made after considering one of the feature variables selected from multiple available variables.The selection of feature variable is done in two steps; firstly by choosing the optimal split for each feature and secondly by taking the feature that has the ability to minimize error. Given N is the total number of samples in the tree; the tree error is calculated on the basis of the weighted average of errors at each node as shown in the following equation.

$$MSE_{tree} = 1/N \sum_{tree} \sum_{D_l} (y_l - y_i)^2 \tag{3.2}$$

MSE is used here to find the difference between trees with and without splits. The largest difference value refers to the optimal split value. This is shown in equation 2.6, where t is the unsplit parent node and t_1 is the branch for which the split was True and t_r the branch for which the split was False.
The peak valued split is calculated by taking the highest subtracted result of the MSE values of the tree with split and without split.The following equation expresses the finding where t is the unsplit parent node and t_1 is the branch for which the split was True and t_r the branch for which the split was False.

$$\Delta.MSE = MSE(t) - n_{t1}/n_{tree} MSE(t_1) - n_{tr}/n_{tree} MSE(t_r) \tag{3.3}$$

Decision trees tend to show the nature of having overfitting problems. Overfitting occurs as as a tree keeps splitting itself until one sample is left at each end node causing increased error percentage. We can resolve the issue in two ways; the first is through restricting the decision tree to have a certain depth, and the second is after we let the entire tree to grow, use a validation set to prune split nodes by finding the nodes that cause increase in errors. In this study we have applied the method discussed in Breiman [12] named 'Bootstrap Aggregation (bagging) Approach' to avoid overfitting problems by taking distinct training set for each tree. By using bagging method we take random samples from all the training data for each decision

tree. Here the error for each tree is computed by using the data that were not present in the training sets (out of bag or OOB samples).
Random forest represents the concept of having many decision trees with random sampling of the training data as well as important features for the purpose of prediction. The core concept here is, many trees together yield more positive outcome in predicting a result compared to only a single tree as mentioned in Breiman [16]. To get the best performance we can tune the following parameters such as the number of trees, features tested at each split and training samples per tree.
By tuning the number of trees, we can control the number of random samples of training data. It has been found that, increased number of tree result in better performance and has less error values. The number of feature tested at each split is to see if the optimal feature is used at the split. By decreasing this number we can ensure that different trees in the forest split in different ways to add diversity in the procedure. The amount of correlation between trees is controlled by the number of training samples per tree. The chances of trees having overlapped training set increases with the increase in the number of training samples per tree [41].

3.2 K Nearest Neighbor

K-nearest neighbors algorithm (k-NN) is a machine learning algorithm used for both classification and regression. It is a simple algorithm that stores all the available cases and models the new given input or case on a similarity measure. The K in KNN denotes the number of nearest neighbor that are used to predict the new data. It is a hyper parameter that should be picked in order to get the best possible fit for the data set. Hyper parameters are parameters that can not be learnt directly from the regular training process. K actually controls the shape of the decision boundary. In case of regression analysis the new data is labeled based on the average value of K nearest neighbors. In contradistinction to statistical approaches that try to distinguish a model from the obtainable data, the KNN method uses the training set as the model.
Any KNN algorithm is specified by points such as the number of neighbors, type of distance used, etc. While using this particular machine learning algorithm there are various parameters that can be used. With the correct usage of these parameters we can certainly improve accuracy. In real world scenario just like our data we can not apply trial and error methodology meaning we can not test with all the possible parameters and identify which one performs better. So, we use optimal parameter tuning to select appropriate parameters. We can discuss some of the important parameters that highly effect accuracy of a KNN model. K is regarded as one of the most important factors of the model that can strongly inuence the quality of predictions. For any given problem domain, a smaller value of k can show a monumental variance in predicted results. On the other hand, a large value of k will conduct a vast model bias. So, we need to choose an appropriate value for k which is neither too large nor too small.
By using the reference of figure 3.1 we can see how k parameter chooses the training set and predicts unknown value. Our goal is to identify appropriate value of k using cross validation to build an optimal model.

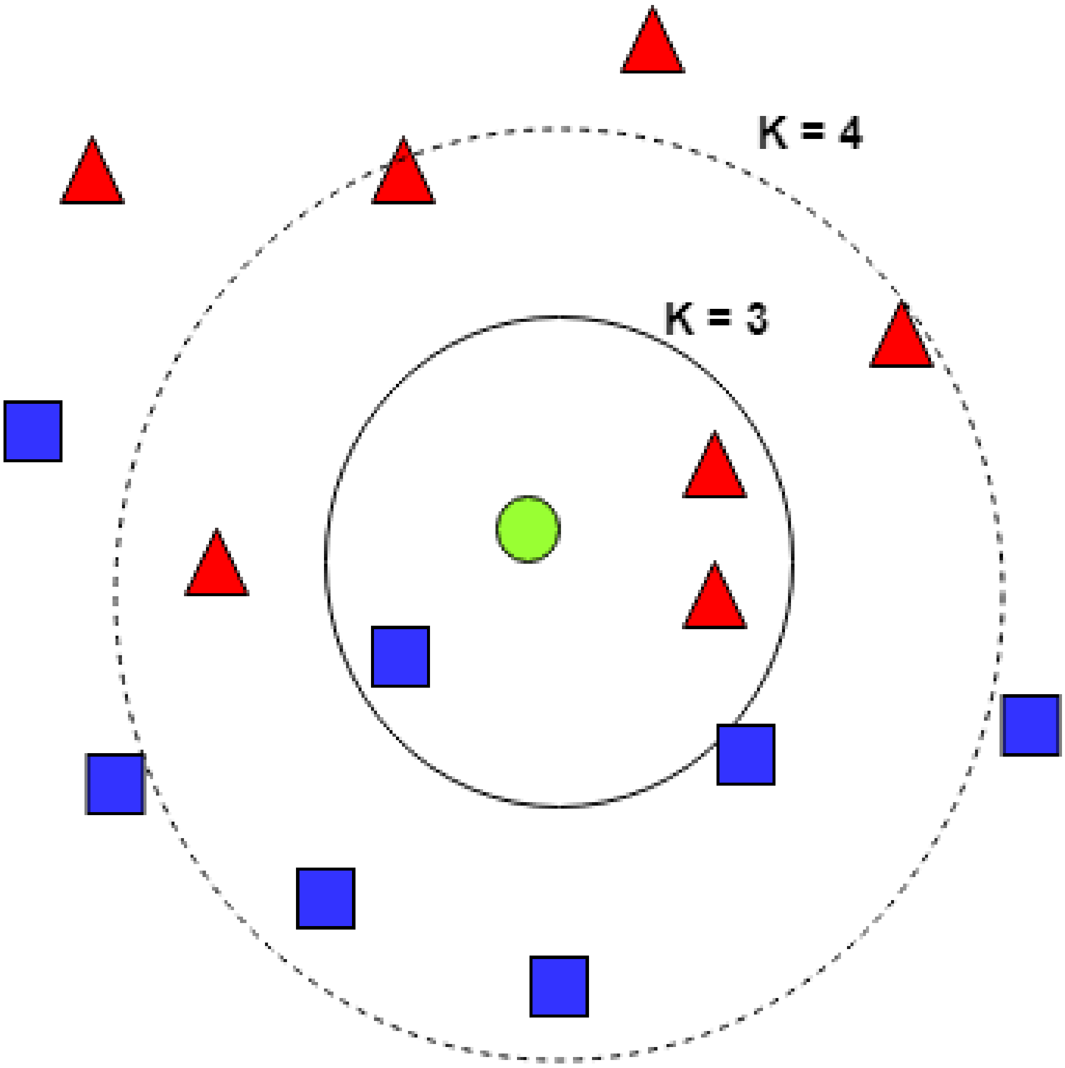

Figure 3.1: Working principle of k parameter

Weights

This function is one of the hyper parameters that effects the prediction precision accuracy. There are many feasible values that could be used. A few of the commonly used weight functions are described below:
• Distance : In this weight distribution, weight is pointed by the inverse of their distance. Closer neighbors of a given point will have a higher influence than neighbors which are further away while calculating weight in this case.
• Uniform : In uniform weight distribution, all points in each neighborhood are weighted equally.
• There is another way to define weight value. This one is a callable, user-defined function which acquire an array of distances then it returns an array of the same shape containing the weights.

Distance Metrics

It is the benchmark used to compute distance. Any metric from scikit-learn or scipy.spatial distance can be used. Some of the commonly used valid values are:
• 'cityblock', 'euclidean', 'manhattan' from the scikit-learn libary
• 'correlation', 'dice', 'hamming', 'minkowski', 'yule' from scipy.spatial
• This could also be a callable function. When callable, it is called on each pair of rows and the resulting value is recorded. This function take two arrays as input and return one value resulting the distance between them.
In the table 3.1 different distant metrics and there characteristics are showed. An-

Identifier	Class Name	Distance Function
"euclidean"	Euclidean Distance	$\sqrt{\sum(x-y)^2}$
"manhattan"	Manhattan Distance	$\sum(\lvert x-y\rvert)$
"cityblock"	Cityblock	—x-y—
"hamming"	Hamming	$\sum_{i=1}^{k}(\lvert x_i-y_i\rvert)$

Table 3.1: Different types of distance metrics

other distance that might be used is 'minkowski'. The Minkowski distance between two variabes X and Y is defined as:

$$D(x_i, x_j) = \sum_{l=1}^{d}((|x_{il} - x_{jl}|)^p) \tag{3.4}$$

Optimal Tuning Parameters

There are various ways to tune parameters in order to control their behavior.

K-fold cross-validation

Main steps to apply cross-validation is to split data set into K "folds" of equal size where each fold acts as the testing set 1 time, and acts as the training set K-1 times. Cross-validated performance is used to judge the out-of-sample performance. Benefits of cross-validation covers more genuine estimation of out-of-sample performance than train/test split. It also reduce variance of a single trial of a train/test split. This tuning can be used for selecting tuning parameters, choosing between models and selecting features. Disadvantage of cross-validation is that it can be computationally expensive, mostly when the data set is large and the model is slow to train.

Cross Validation Score

We can also use cross-validation to estimate the test error associated with a learning method in order to evaluate its performance. It can also be used to select the appropriate level of exibility. It is a well established technique that is used to obtain estimates of unknown model parameters.

GridSearchCV

This is a more efficient parameter tuning which allows you to define a grid of parameters where parameters will be searched using K-fold cross-validation. This is like an self-regulating "for loop" of the above function. First the parameter values that should be searched are defined then a parameter grid is created to map the parameter names to the values that should be searched.

3.3 Recurrent Neural Network

Recurrent Neural Network or popularly known as RNN are neural networks designed for capturing information from time series data. They can take variable size input and give variable size output which works really well for time series data. Understanding RNN could be tricky a lot of representation leads to misconception. RNN works with the following recursive formula:

$$S_t = F_w(S_t - 1, X_t) \tag{3.5}$$

The new state of the recurrent neural network at time t is a function of it's old state at time t-1 and the input at time t. S_t = current state(at time t) $S_t - 1$ = previous state X_t = input at time t This function is the basic idea behind RNN. The simplest implementation of RNN is showed in figure 3.2
The recursive function is a $tanh$ function where we multiply the input state with weight of input W_x and the previous state with W_s and pass it through a $tanh$ activation which gives us the new state. Now to get the output vector we multiply the new state with that is St with W_y.

$$S_t = tanh(W_s S_t - 1 + W_x X_t) Y_t = W_y S_t \tag{3.6}$$

$$Y_t = W_y S_t \tag{3.7}$$

If we unroll the RNN following the figure 3.3 we can see that there is previous state S_0, input at time step 1 is X_1 , these goes 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)

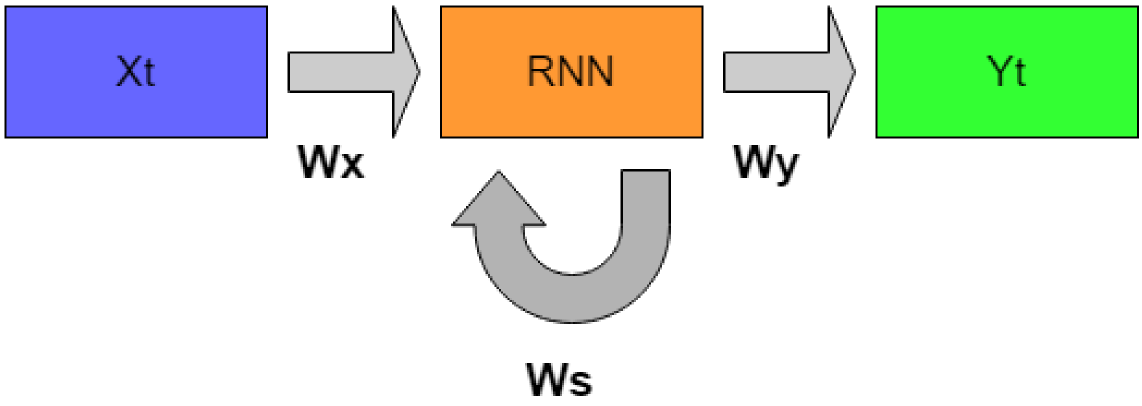

Figure 3.2: Simple RNN

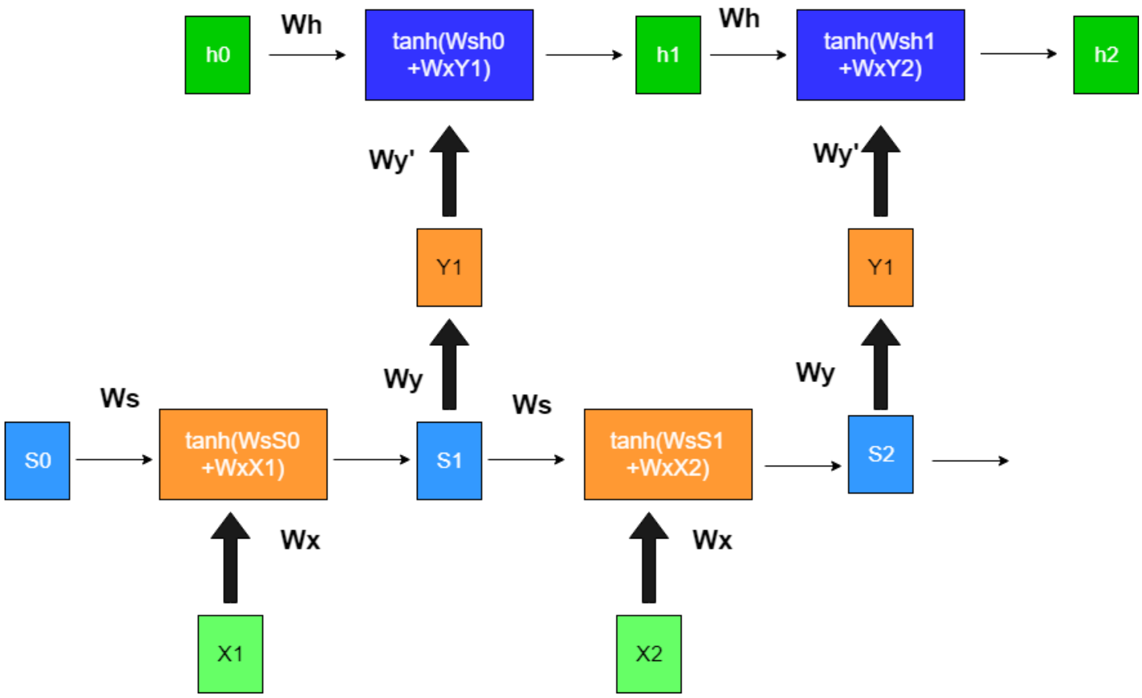

Figure 3.3: Simple RNN Unrolled

and to get output we multiply S_1 with W_y. Now the new state S_1 and input X_2 is the input for the new time step. We again get S_2 and output by multiplying it with W_y. But the problem is we used the same set of weight throughout the model. In case of multilayer RNN we serve outputs as input of our next layer. Here Y_1 and Y_2 act as input in the next layer. In general deeper networks give better accuracy but in RNN we do not go a lot deeper. People generally use 2 to 3 layer deep modules. RNN learn using back propagation through time. So, we calculate the loss using the output and go back to each state to update the weights by multiplying gradient. But the update in weight is almost zero which is negligible. That means our model would not learn anything. This called the vanishing gradient problem. To overcome this problem and improve accuracy we add more interactions to RNN. Which is the idea behind LSTM.

Long Short Term Memory (LSTM)

LSTM solves the vanishing gradient problem. LSTM cells can address this issue by incorporating memory cells in the hidden layer of RNN. For the above reason LSTM has gained popularity in predicting time series data. In a paper by [53] they applied LSTM on a realistic data set, the experimental results of which show that the proposed method has higher forecasting accuracy and applicability, as compared with existing methods.

$$F_t = (\sigma(W_f S_t - 1 + W_f X_t)) \tag{3.8}$$

$$i_t = (\sigma(W_i S_t - 1 + W_i X_t)) \tag{3.9}$$

$$o_t = \sigma(W_o S_t - 1 + W_o X_t) \tag{3.10}$$

$$C'_t = tanh(WcS_t - 1 + W_c X_t) \tag{3.11}$$

$$Ct = (i_t*, C'_t) + (f_t * C_t - 1) \tag{3.12}$$

$$h_t = o_t * tanh(C_t) \tag{3.13}$$

In the above equations the variables represents the following meanings - F_t=Forget gate, i_t=Input gate, o_t=Output gate, C'_t=Intermediate cell state, Ct=Cell state and h_t=New State Advantage of LSTM is that it has three gates. These gates and cell states are additional interactions. Where we have the forget gate which takes the old state and the input and multiplies it with respective weights. Then we pass it through a sigmoid activation. We have input gate and output gate and the same mechanism works through them. The important point is each gate has different sets of weight. Again, C' is an intermediate cell state using which we calculate the cell state. Then we get the new state by multiplying tanh activation of the cell state with output gate. The process is can be visualized by figure 3.4.

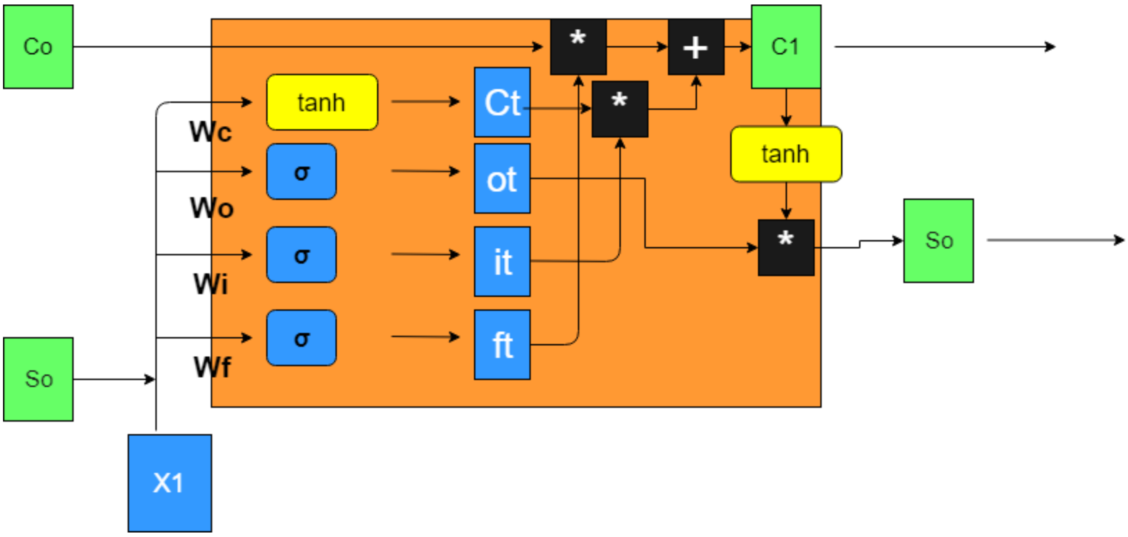

Figure 3.4: LSTM Mechanism

Chapter 4

System Implementation and Result Analysis

Our goal is to observe how machine learning approach aid in the daily forecast of power consumption based on the historical data as well as find hidden patterns and their effect in forecast result. With that goal in mind, we have experimented on the historical dataset of three years (June, 2015 – June, 2018) using three popular computational intelligence techniques - Random Forest (RF), K Nearest Neighbor (KNN), and Long Short Term Memory (LSTM).

The following sections elaborately discuss how the dataset was assembled and the training process of Short-term Load Forecasting (STLF) system using the above mentioned algorithms. The last section shows the in depth result analysis along with a comparative study of the test results.

4.1 Data Processing

This section highlights the collection and cleaning process of the data used in the experiments. In section 4.1.1, the methodologies used to obtain the data are explained. Section 4.1.2 discusses actions needed to pre-process and filter the raw data with exclusion of noisy and misleading data. The last part of this section explains some preliminary data analysis along with inspection of influential features for this research.

4.1.1 Data Collection

In order to forecast demand in any circumstance, an informative dataset with necessary features is a must. At the preliminary stage of the research it was found that such compact dataset containing essential features to predict the daily amount of electricity consumption in Bangladesh is not available in the availing online resources. The centrally statutory body, Bangladesh Power and Development Board (BPDB) is responsible for serving as one of the main integrated utilities to monitor the generation, transmission and distribution of power in Bangladesh.

BPDB hosts the nationwide daily power generation archive reports (in PDF format shown in Fig. 4.1) from 2009 to present day [54]. To facilitate our experiments a program was developed with the target to convert these scattered data into a functional dataset. Fig. 4.2 illustrates the first twenty entries from the dataset and the flowchart in Fig. 4.3 represents the self-devised program for the creation of the dataset.

(D)	Actual data of 01.06.15 (Yesterday)		Monday	:	
01.	Max. Demand (Generation end)	:	7782.00	MW, at =	20:00 hrs
02.	Max. Demand (Sub-station end)	:	7206.00	MW, at =	20:00 hrs
03.	Highest Generation (Generation end)	:	7782.00	MW, at =	20:00 hrs
04.	Minimum Generation (Generation end)	:	5838.00	MW, at =	8:00 hrs
05.	Day-peak Generation (Generation end)	:	6338.00	MW, at =	12:00 hrs
06.	Evening-peak Generation (Generation end)	:	7782.00	MW, at =	20:00 hrs
07.	Evening Peak Load-shed (Sub-station end)	:	0.00	MW, at =	20:00 hrs
08.	Generation shortfall at evening peak due to :				
	a) Gas limitation	:	290	MW	
	b) Low water level in Kaptai lake	:	119	MW	
	c) Plants under shut down/ maintenance	:	1580	MW	
09.	Total Energy (Generation + India Import)	:	157.65	MKWh	
	By Gas = 102.01 MKWH		By Oil = 39.11 MKWh		
	By Coal = 4.09 MKWH		By Hydro = 1.69 MKWh		
10.	Total Gas Supplied	:	1003.48	MMCFD	

11.	Zone wise Demand and Load-shed at Evening Peak (Sub-station end) :						
Zone	Demand MW	Supply MW	Load Shed MW	Zone	Demand MW	Supply MW	Load Shed MW
Dhaka	3048	3048	0	Mymensingh	463	463	0
Chittagong	736	736	0	Sylhet	259	259	0
Khulna	828	828	0	Barisal	141	141	0
Rajshahi	732	732	0	Rangpur	374	374	0
Comilla	625	625	0	Total	7206	7206	0
12.	Fuel cost :	(a) Gas =	68876517 Taka	(c) Coal =	15937106		Taka
		(b) Oil =	495897402 Taka	Total =	580711024		Taka
13.	Maximum Temperature in Dhaka was :		36° C				
14.	Export through East-West interconnections :						
	At evening peak-hour	:	-300	MW, at	20:00 hrs		
	Maximum	:	40	MW, at	15:00 hrs		
	Energy	:	7.8675	MKWh			

Figure 4.1: Sample of daily power generation report (in pdf format) from BPDBs website

	date	max_demand_gen	highest_gen	min_gen	day_peak_gen	eve_peak_gen	eve_peal_load_shedding	water_level_kaptai	max_temp	total_gas	total_energy
0	31.05.15	7817.00000	7817.00000	5623.00000	6529.00000	7817.00000	0.00000	78.99000	35.50000	1007.95000	157.20000
1	01.06.15	7782.00000	7782.00000	5838.00000	6338.00000	7782.00000	0.00000	78.95000	36.00000	1003.48000	157.65000
2	02.06.15	6869.00000	6869.00000	5167.00000	6482.00000	6869.00000	0.00000	78.85000	36.00000	1004.19000	148.74000
3	03.06.15	7549.00000	7558.00000	4975.00000	5658.00000	7549.00000	0.00000	78.50000	35.70000	1037.43000	146.43000
4	04.06.15	7676.00000	7752.00000	6017.00000	6552.00000	7676.00000	0.00000	78.18000	36.20000	1049.77000	163.47000
5	05.06.15	7312.00000	7444.00000	5841.00000	6195.00000	7312.00000	0.00000	77.90000	36.50000	1029.04000	165.22000
6	06.06.15	7372.00000	7435.00000	6060.00000	6613.00000	7372.00000	0.00000	77.66000	34.80000	1029.04000	161.25000
7	07.06.15	7496.00000	7496.00000	6092.00000	6861.00000	7496.00000	0.00000	77.36000	35.60000	1015.44000	170.66000
8	08.06.15	7573.00000	7704.00000	6197.00000	6656.00000	7573.00000	0.00000	76.90000	36.00000	1005.77000	164.10000
9	09.06.15	7653.00000	7653.00000	5957.00000	6840.00000	7653.00000	0.00000	76.28000	35.00000	1028.93000	165.19000
10	10.06.15	7385.00000	7385.00000	5843.00000	6036.00000	7385.00000	0.00000	75.36000	31.20000	959.16000	155.58000
11	13.06.15	6966.00000	6966.00000	4476.00000	4476.00000	6966.00000	0.00000	74.34000	31.60000	921.74000	137.70000
12	14.06.15	6869.00000	6869.00000	4722.00000	5504.00000	6869.00000	0.00000	74.59000	32.50000	960.05000	139.72000
13	15.06.15	7200.00000	7208.00000	5142.00000	6019.00000	7200.00000	0.00000	74.70000	31.60000	985.15000	147.20000
14	16.06.15	7300.00000	7085.00000	4988.00000	6018.00000	7085.00000	194.00000	74.78000	31.60000	959.24000	146.31000
15	17.06.15	7297.00000	7297.00000	5431.00000	6051.00000	7297.00000	0.00000	74.87000	32.60000	996.82000	149.24000
16	18.06.15	7500.00000	7348.00000	5586.00000	6184.00000	7279.00000	200.00000	74.77000	33.60000	998.10000	151.16000
17	19.06.15	7650.00000	7407.00000	5627.00000	5685.00000	7407.00000	219.00000	74.41000	35.00000	989.84000	153.51000
18	20.06.15	7700.00000	7407.00000	5992.00000	6265.00000	7407.00000	264.00000	74.15000	34.20000	1020.69000	157.40000
19	21.06.15	7600.00000	7472.00000	5964.00000	6272.00000	7397.00000	183.00000	74.24000	33.00000	1041.85000	158.51000
20	22.06.15	7365.00000	7365.00000	5420.00000	5845.00000	7365.00000	0.00000	74.37000	29.30000	1039.36000	148.64000

Figure 4.2: Sample entries of the dataset

Initially the reports from the archive are accessed and downloaded from the BPDB website using report numbers which are unique to their corresponding dates. The data mining program consists of two sub units – a 'Data Extractor Program' which uses regular expressions to parse the PDF formatted files to extract specific information and a CSV Loader Program' that transfers the extracted data (appended in a list) into a CSV file which serves as the base dataset.
Due to inconsistency in the patterns in which the reports were documented in different years as well as chunk of missing reports, the data extracted follows the period of June, 2015 to June 2018 marking a consistent dataset which holds essential information from daily reports of mentioned three years for the forecasting purpose. Table 4.1 shows the table which explains the features of the dataset. The next section will explain how the extricated data was used to create a clean dataset on which the experiments are conducted later.

4.1.2 Data Cleaning

The dataset contains data for the period of three years (01-06-2015 to 10-06-2018). This ultimately totaled to 1126 rows in the dataset. Measurement of noise influences on collected data is a crucial phenomenon before starting any experiments. Unlike any other database generated from raw data, this dataset also contained some erroneous entries.

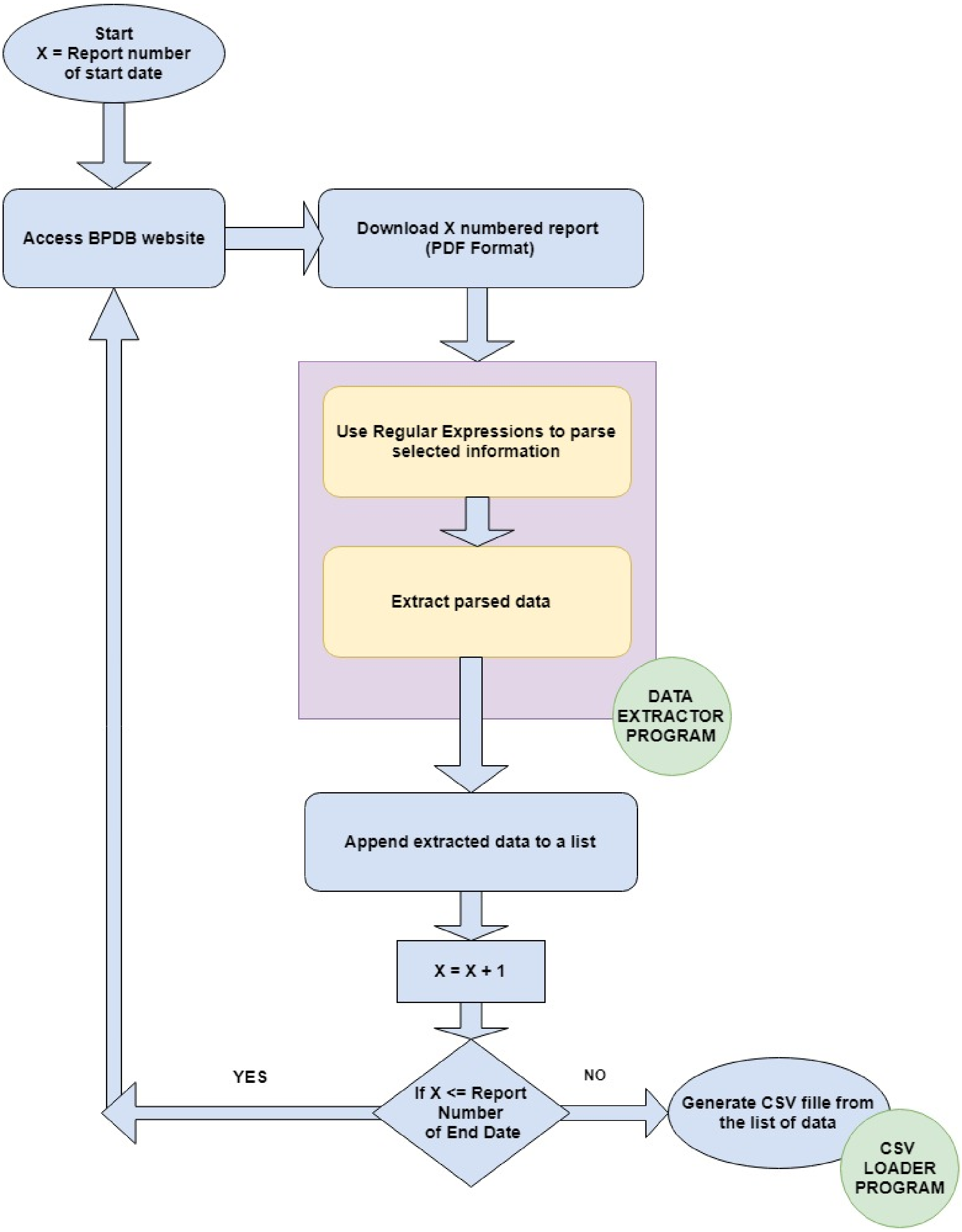

Figure 4.3: Data extraction process

Column name	Unit	Description
date type:Timestamp	Date-time	The date of the period for which this row holds information
max_demand dtype:float64	MW	Highest load of the power system under the specific amount of timestamp
highest_generation dtype:float64	MW	Highest load generated under a particular amount of datetime
Min_gen type:float64	MW	Minimum load generated under a particular amount of datetime
day_peak_gen type:float64	MW	Generation at peak hour during day
eve_peak_gen type:float64	MW	Generation at peak hour during night
eve_peak_load_shedding dtype:float64	MW	Load-shedding at peak hour during night
water_level_kaptai dtype:float64	ft	The minimum height to generate electricity from water
max_temp type:float64	°C	Maximum temperature of the particular day
total_gas type: float64	MMCFD	Total gas supply on the particular day
total_energy (Generation + India Import) dtype: float64	MKWh	Total energy consumed on the particular day

Table 4.1: Relevant fields in the dataset

Initially all of the available daily generation reports in the BPDBs website were seek to be accessed to extricate necessary data. But due to the process of manual generation of reports, many nonsensical entries and various patterns of documentation throughout different years continuously disrupted the unified data collection process using regular expressions. The final data was adopted starting from June 1st 2015 to 30th June 2018 as the reports from these dates had consistent pattern of documentation and regular entries.

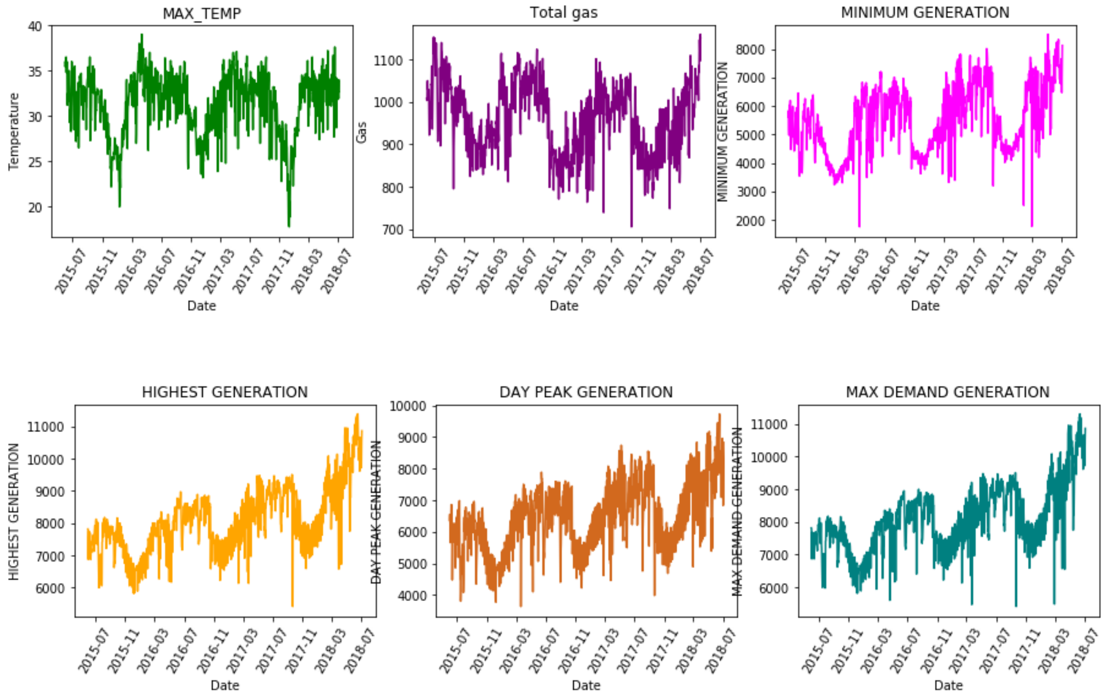

Figure 4.4: Some plots of different features against date

Secondly, the case was found where the dates in the extracted time series data did not have the standard DateTime format. To resolve this, properly formatted dates were generated to replace the inaccurate ones for the purpose of training the machine learning models accurately.
A thorough check of the data set was executed to locate any missing rows, duplicate values along with null and missing values. The data set did not contain any missing rows; duplicate values were terminated and null and missing values were replaced with a self-devised algorithm.
The algorithm is implemented for each of the columns. Each row with date was checked against each column to find any null or zero values. If null or zero values were found; we replaced that by taking the mean value of all the other values with similar time period found in the dataset for that respective column. For the column 'eve_peak_load_shedding' this process was not implied as most of the cells contain zeroes referring to no load shedding.
A final filtering was done to ensure that the columns had no outliers or missing values as shown in Fig. 4.4 and FIg. 4.5 by visualizing the data through plotting the data for each column.

4.1.3 Load Data Analysis

The basic step towards scientifically experimenting with any available data is through conscientious analysis of the properties of the data. On the basis of previous researches, some preliminary analysis were done on our dataset as well as due to the periodic nature of the data, some frequency analysis were also conducted to detect commonness in features.

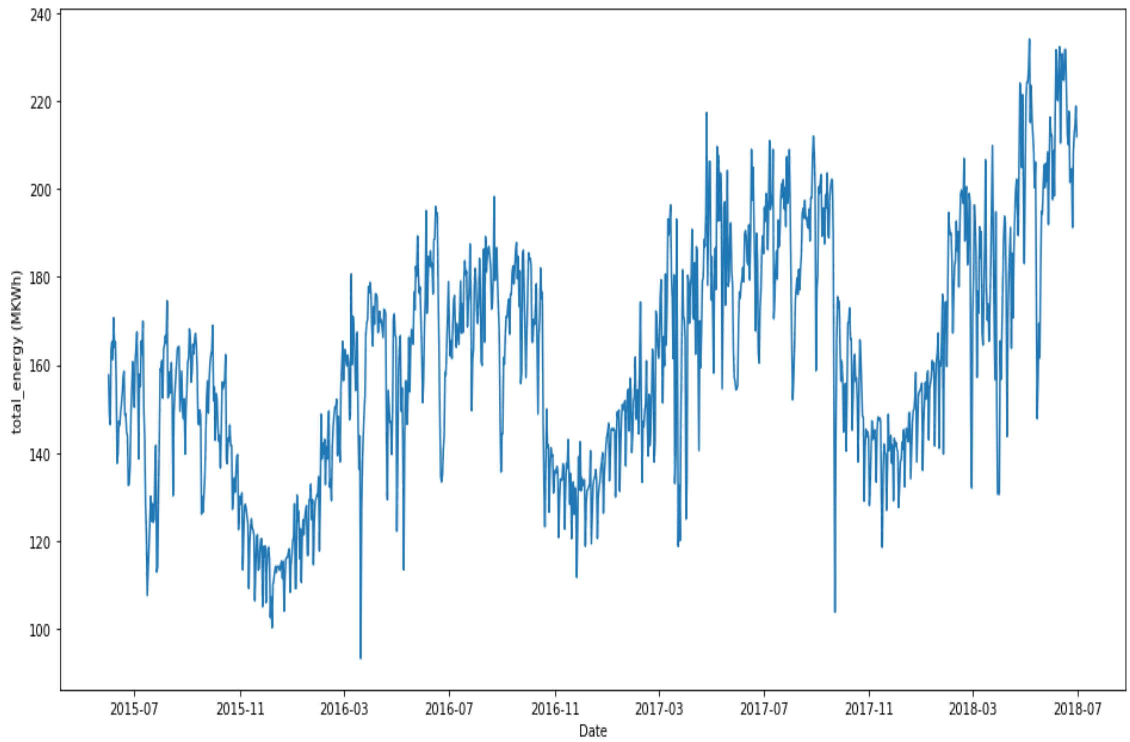

Figure 4.5: Total power consumption of 3 years (June, 2015 – June, 2018)

The time factors in the data set initially only included the time of the year, but later the day of the week was computed against each row. The load between weekdays and weekends have significant differences and these differences play vital roles in load forecasting according to Mu et al. [31]. The load distribution of different weekdays also behave differently. According to the historical load study of out data in Fig. 4.6, we can see that Wednesdays and Thursdays being adjacent to weekends have structurally different loads than Sunday through Tuesday. The demand drops on weekends, particularly on the second day of the weekend, Saturday, and then steeply rises on the first workday, Sunday and onward.
The historical study of load demand according to the months behave as expected, the power consumption rate being significantly greater in summer than winter. Load forecasting during summer and winter seasons can be improved greatly if temperature sensitivity is considered [38]. We can see in Fig. 4.7 that the power consumption rates are highest during the summer season (April - July) while during the winter season (October - January) the consumption rate drastically reduces.
Weather conditions have tremendous influence in the load forecasting. Many researches done on short term load forecasting deduct that, forecast weather variables are perhaps the most important factors in influencing forecast result. For load forecasting, multiple weather parameters can be selected. Two of the most commonly

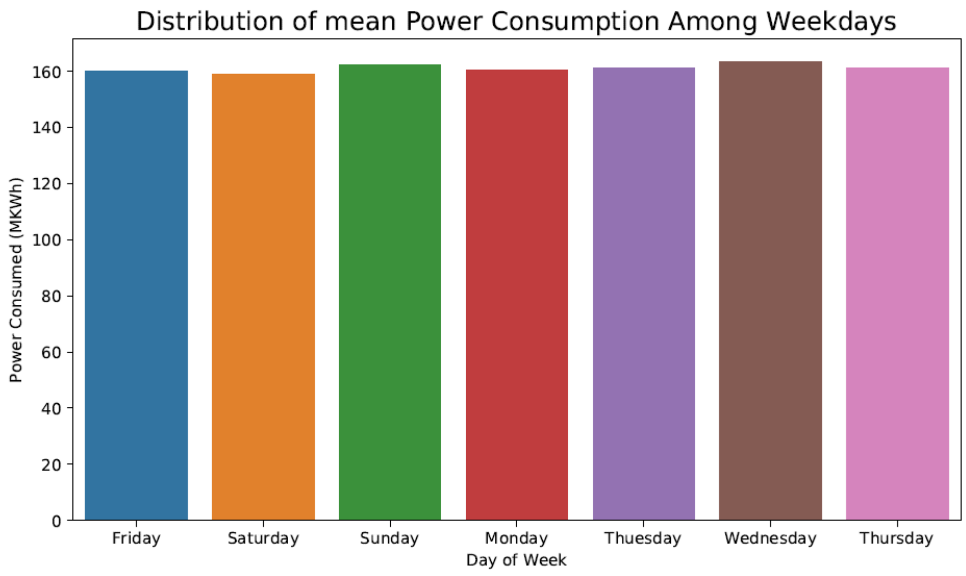

Figure 4.6: Distribution of mean power consumption among weekdays

used weather variables are temperature and humidity. Meng, Xiao Fang et al. [45] stated in their work that the accuracy of load forecasting is highly dependent on temperature and the effect of temperature in predicting short term loads is more significant than any other features. An electric load prediction survey published by Hippert et al. [17] indicated that of the 22 research reports considered, 13 made use of temperature only, 3 made use of temperature and humidity, 3 utilized additional weather parameters, and 3 used only load parameters. For this data set the maximum temperature is used to find the association between load data and weather. The inherent correlation between load and temperature can be seen in Fig. 4.8 which exemplifies the piece wise linear relationship of the mentioned factors.
According to a survey [55], in 2011, over 2000 millions of cubic feet of gas per day (MMCFD) were produced from 79 natural gas wells found in 23 operation gas fields of Bangladesh. This is an important factor because more than three-quarters of the nation's commercial energy demand is being met by natural gas [40]. This influential sector caters for around 40% of the power plant feed-stock, 17% of industries, 15% captive power, 11% for domestic and household usage, another 11% for fertilizers, 5% in compressed natural gas (CNG) activities and 1% for commercial and agricultural uses [55]. Fig. 4.9 highlights the linear relationship between the consumption of gas to produce electricity and load data.
The correlation matrix in Fig. 4.10 visualizes the heat map of the performance comparison of different features. The "heat" in heat map refers to the amount of action on a particular area and the warmer the color the more actions are happening. It can be observed that along with time of the year, temperature and amount of gas consumed; Minimum Generation (Generation end), Day-peak Generation (Generation end), Evening-peak Generation (Generation end), Highest Generation (Generation end) and Max. Demand (Generation end) also have potential influences on the load data.

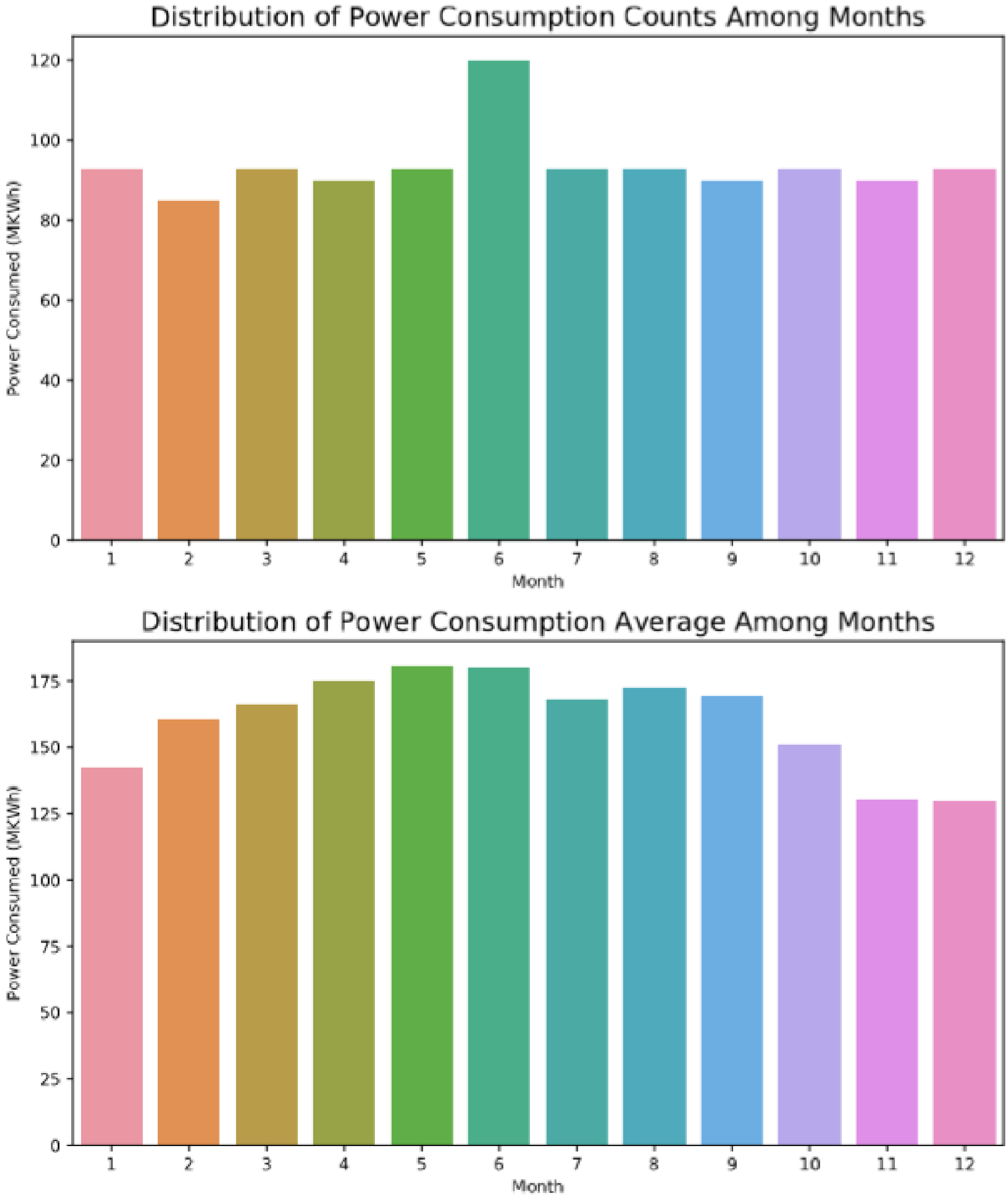

Figure 4.7: Distribution of average power consumption among months

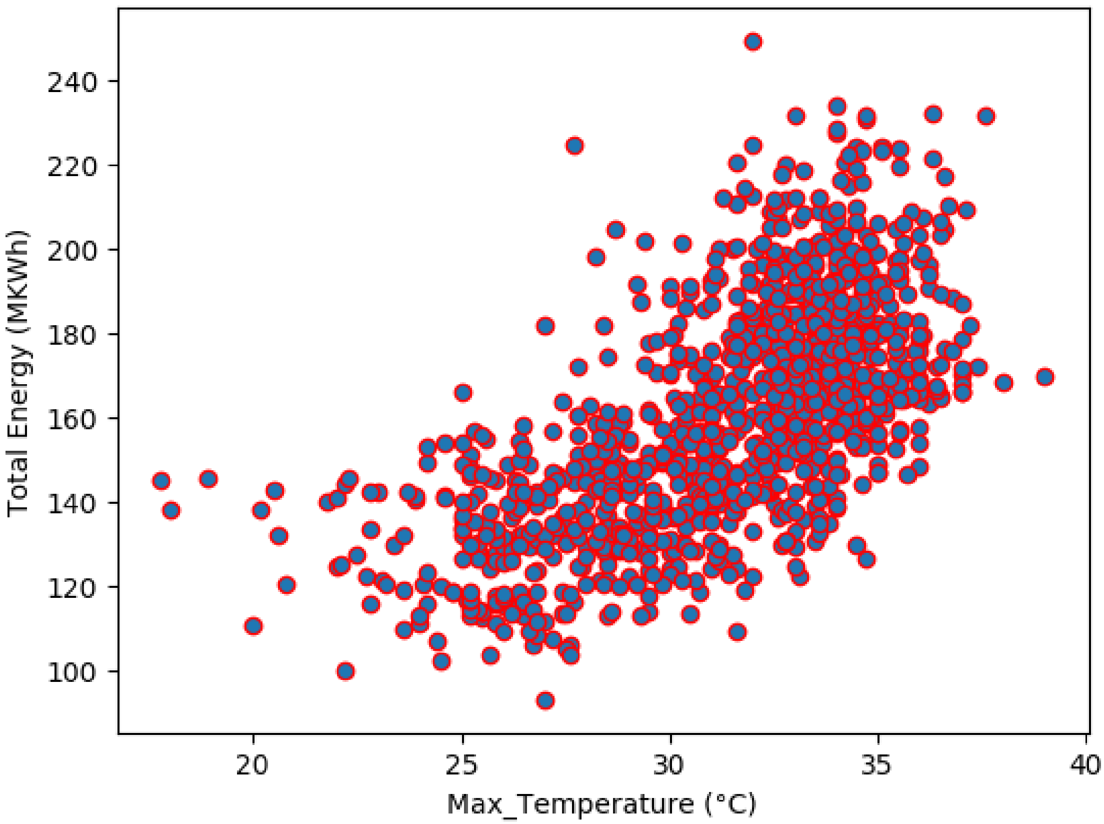

Figure 4.8: Correlation between demand and temperature

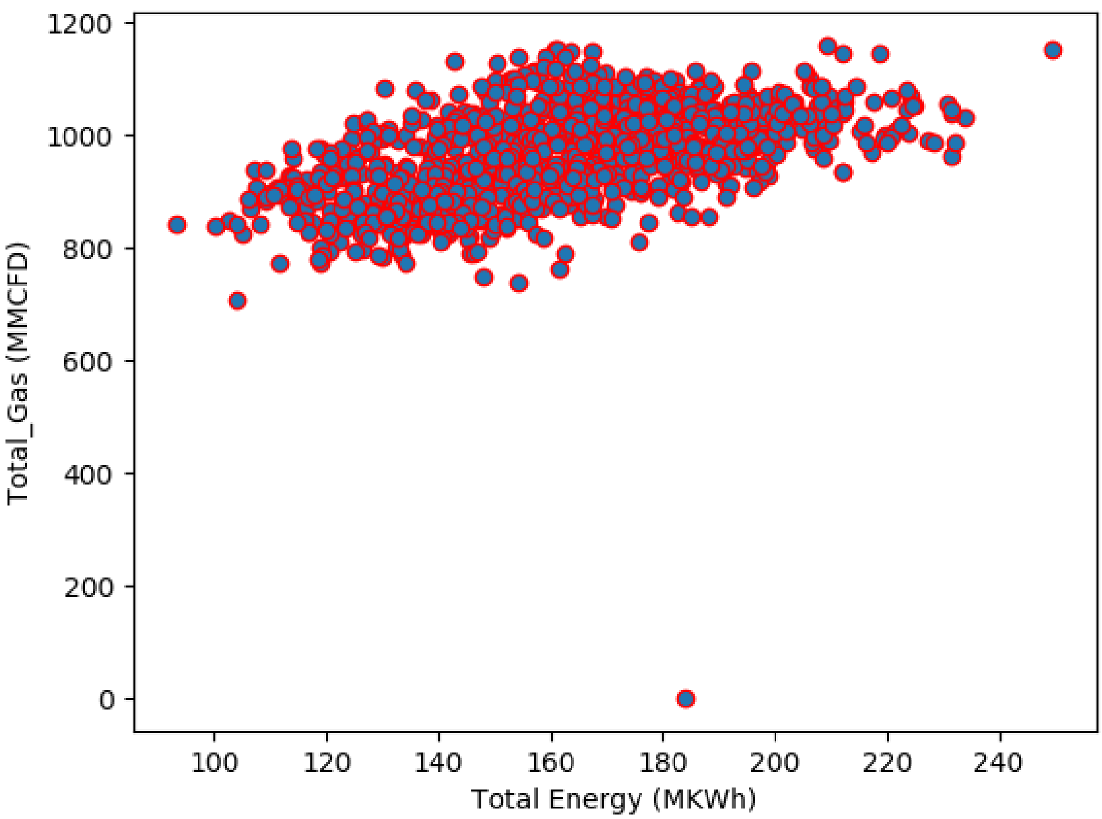

Figure 4.9: Correlation between demand and gas consumption

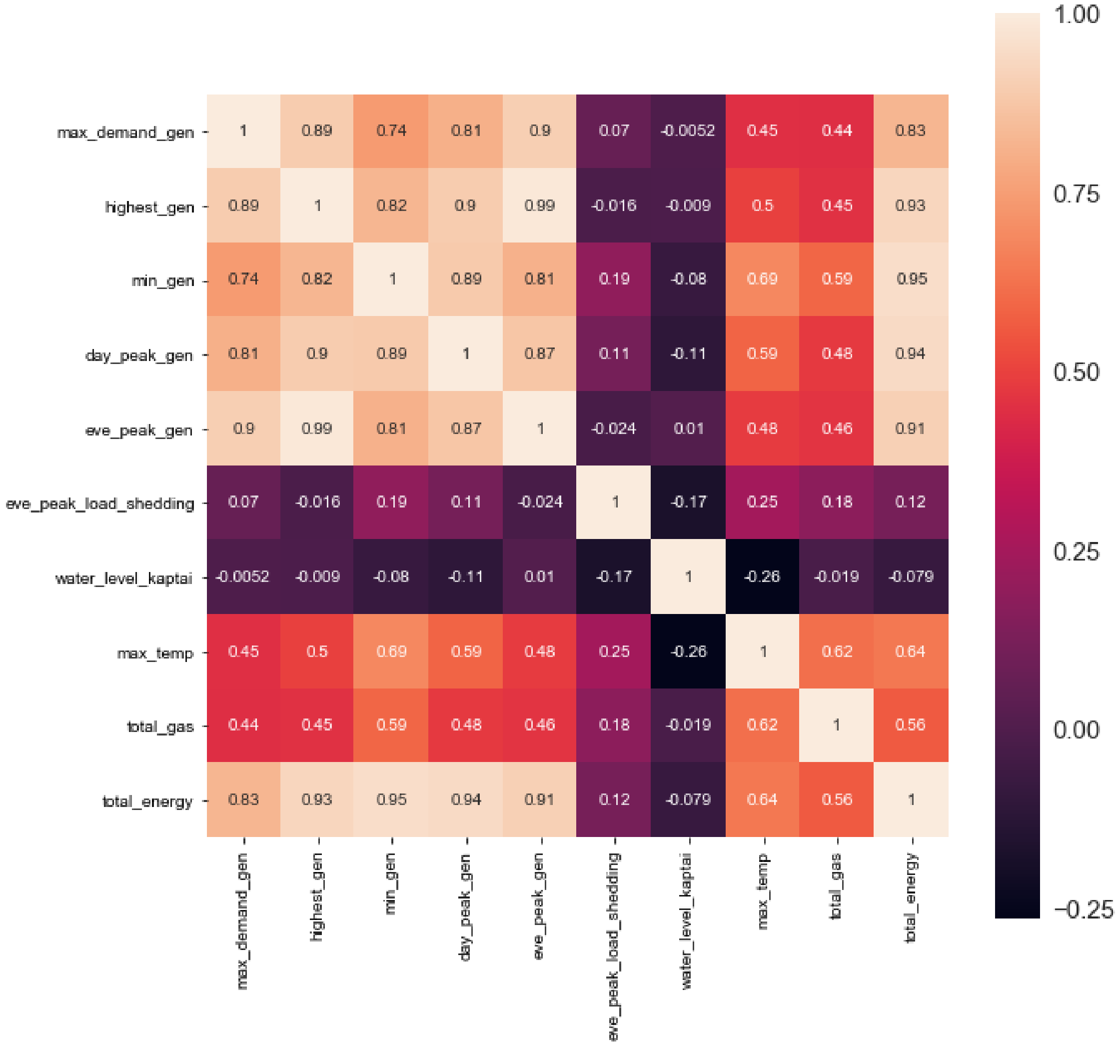

Figure 4.10: Correlation matrix of features

4.2 Training SLTF System

4.2.1 Random Forest

Random forest, first proposed by Tin Kam Ho of Bell Labs in 1995 [10], is an ensemble learning method for classification and regression. The algorithm for inducing a random forest was developed by Leo Breiman and Adele Cutler in 2001 [16]. The main idea behind the algorithm is to build a large number of decision trees (base learners) with the motivation to reduce error correlation between classifiers by using a random selection of features to split on at each node. Rather than just simply averaging the prediction of trees (which we could call a "forest"), the random forest model uses two key concepts that gives it the name random:

1. Random sampling of training data points when building trees.
2. Random subsets of features considered when splitting nodes.

These increase diversity in the forest leading to more robust predictions. When it comes time to make a prediction, the random forest takes an average of all the individual decision tree estimates. The main advantages behind training the STLF system with random forest is that random forests are easy to build and faster to predict with resistance to over training and over-fitting of data; ability to handle data without preprocessing or re scaling and the RF trained model is resistant to outliers and can handle missing values. 4.11 represents the flowchart of basic steps of training our model through random forest implementation.
The first step in training the model starts with separating the data into features and targets. The target, also known as the label, is the value we want to predict (in this case the daily power consumption, found in the column 'total_energy') and the features are rest of all the columns the model uses to make the prediction. The Pandas dataframes in which parts of the dataset are stored are converted to Numpy arrays to implement the algorithm.
The second step is splitting the data into test and train set. During the training session, we will let our model to see the target values, in this case total energy, along with other features. By doing this, under supervised learning, we can ensure that our model is building an understanding to predict the target value (total energy) from the features other than target value. Over the training period the model slowly learns if there are any relationships between the features and the target value. Later, when it comes to evaluate the model, the model no longer has access to the target values and makes predictions based only on the features of the test set. We then compare the predictions with the true values of target from the test set to compute the accuracy of the model.
To train our model we have randomly split our data into train and test sets where the training dataset consists of 75% of randomly chosen data and test data set is made with rest of the 25% data. For this project, We have set the random state to 42 which means the results will be the same each time the split is run for reproducible results. After all the work of data preparation, the random forest regression model is imported from skicit-learn (a powerful machine learning library), instantiated, and fit (scikit-learn's name for training) on the training data to perform predictions and find relationships among the target and features.

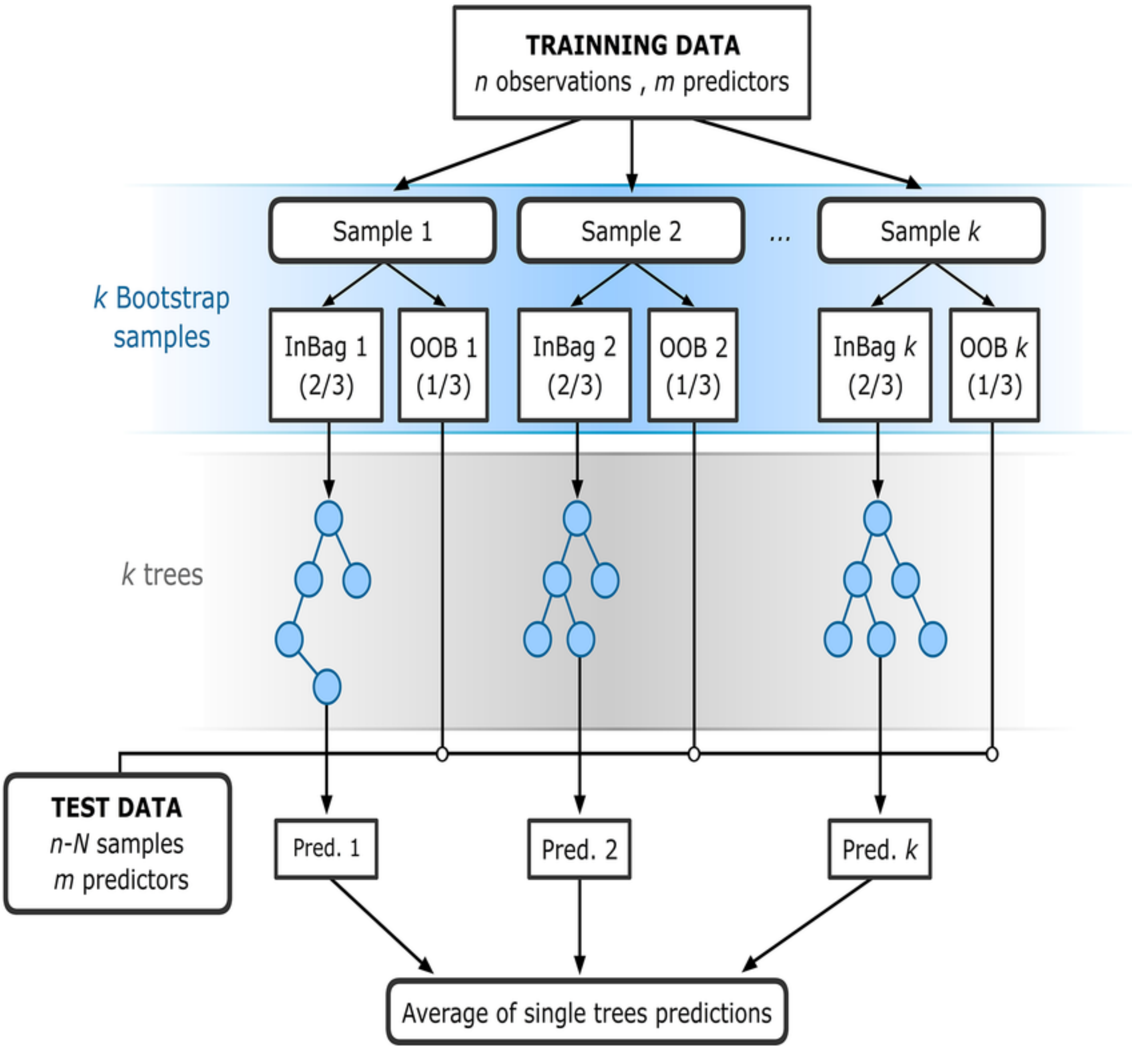

Figure 4.11: Workflow of random forest [46]

4.2.2 K Nearest Neighbor

The purpose of the libraries used is as follows:
• numpy—used to easily make matrix calculations and mathematic manipulations which are essential for any ML model
• pandas—used to define a nice data structure for your training data
• sklearn—a tool used for data analysis (for example normalizing or clustering data)
• matplotlib—used to display our data
We used pandas library to load the .csv file and store it into a DataFrame. We will first map the dates and create a discrete dataset so that the original dataset remain unaltered. This mapping creates many new features such as: 'Year', 'Month', 'Week', 'Day'. We can also introduce many other features that may tend to have a significant effect on the prediction process. For instance, our hypothesis was that the weekend of the week could potentially affect the power consumption as most of the offices and educational institutions remain close on that day. So we mapped our data to identify the day of the week because this could play a significant role in overall electricity consumption.
The first step in training the model starts with separating the data into features and targets. The target is the value we want to predict, in this case the daily power consumption, and the features are rest of all the columns the model uses to make the prediction. As we choose the features in such a way that it contain a significant correspondence with the target column, so the model's job here is to learn this relation. Then while evaluating the model, we make predictions on the testing set where it can predict the target using the features. So, we need to drop the target column which is "total_energy" here. Then we need to split the data into train and validation sets to check the performance of the model. Here I have set the 30% data into test and the other 70% data before that into train.
In k-NN regression, the k-NN algorithm is used for estimating continuous variables. In our algorithm implementation, we used grid search using scikit-learn's GridSearchCV library which stands for Grid Search Cross Validation. By default, the GridSearchCV's cross validation uses '3-fold KFold' or 'Stratified-KFold' depending on the situation. We can cross validate on the entire test data set and then report the CV score as the accuracy.
At first we define our parameter values that should be searched. we defined 10 values in range of 4 to 16. Then a parameter grid is created known as param_grid to map the parameter names to the values that should be searched. This is basically all the different parameters we can have. It is a simple python dictionary. This should be given as key: value syntax. From the given parameters GridSearchCV finds exactly which parameter is best for our model.
Then we instantiate the model and pass the parameters. The model is ready to do 10-fold cross validation on a KNN model using classification accuracy as the evaluation metric. By printing the fitted function we can see the values of the parameters. We can also look for the best value of the parameters using .best_params_ function.

We can see from the above figure that uniform weight, minkowski distance metric and k=8 has found to give most optimal result by GridSearchCV. CV=5 helps to select different train and test data while modeling the algorithm. So, in this model we are looking at 8 different points where the features matches with the features of

```
GridSearchCV(cv=5, error_score='raise-deprecating',
       estimator=KNeighborsRegressor(algorithm='auto', leaf_size=30, metric='minkowski',
          metric_params=None, n_jobs=None, n_neighbors=5, p=2,
          weights='uniform'),
       fit_params=None, iid='warn', n_jobs=None,
       param_grid={'n_neighbors': [4, 5, 6, 7, 8, 9, 12, 14, 16]},
       pre_dispatch='2*n_jobs', refit=True, return_train_score='warn',
       scoring=None, verbose=0)
{'n_neighbors': 8}
```

Figure 4.12: Optimal parameters chosen by GridSearchCV

the target variable we want to predict. It takes the mean of the target variable of the matched features and show that as the predicted value. Based on the independent variables, KNN finds the similarity between new data points and old data points. The algorithm of implementing our KNN model works as shown in Fig. 4.13

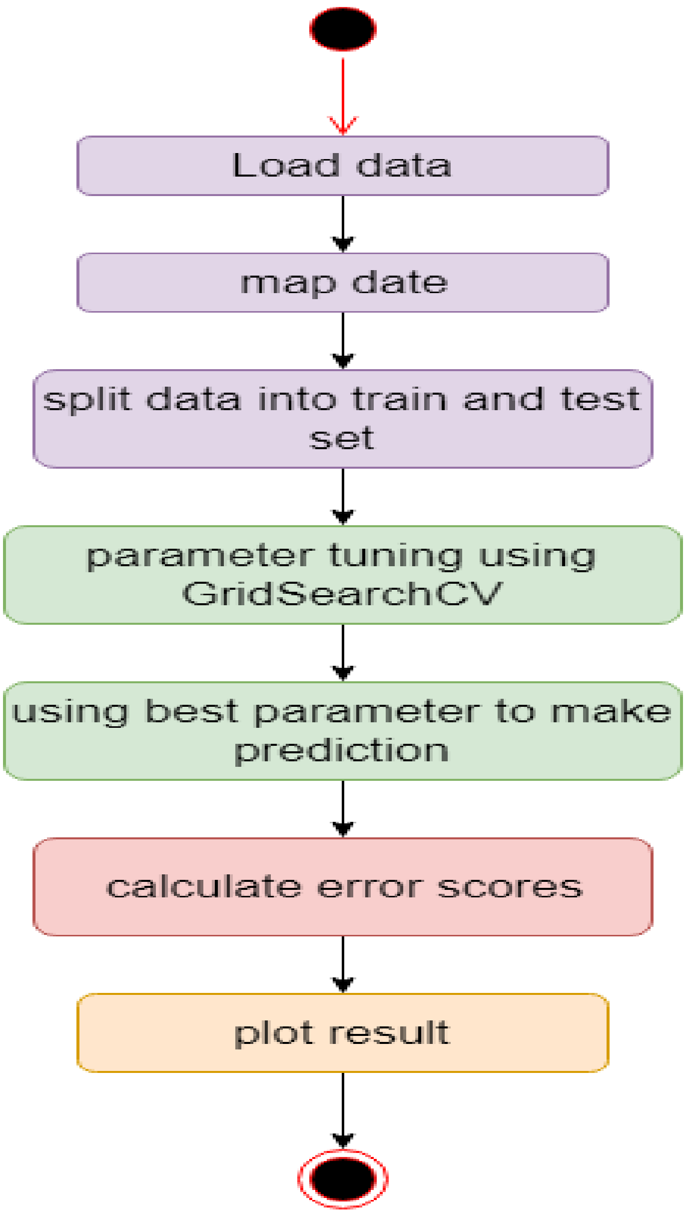

Figure 4.13: Workflow of KNN implementation

4.2.3 Long Short Term Memory

In this section we will discuss the implementation of LSTM network. As we know LSTM has the ability to remember pattern and sequence of data which is mandatory to predict time series data with minimum accuracy. LSTM network can also deal with complexity of order dependence. To build our LSTM model we used Python 3.6 Python framework. Pandas library was used to read dataset from our CSV file. Numpy and Preprocessing from Sklearn was used normalize and again denormalize to visualize the result. We used Keras which is a highlevel neural networks API written in Python and capable of running on top of TensorFlow. Keras is a deep learning library that allows both convolutional networks and recurrent networks, as well as combinations of the two. It also runs seamlessly on CPU and GPU.
After normalizing the data we build a method to load data where we pass our dataframe and window size. Our data is in 2D matrix consisting columns and corresponding rows. But input to a LSTM network is 3D matrix. The other dimension of this matrix is time steps.

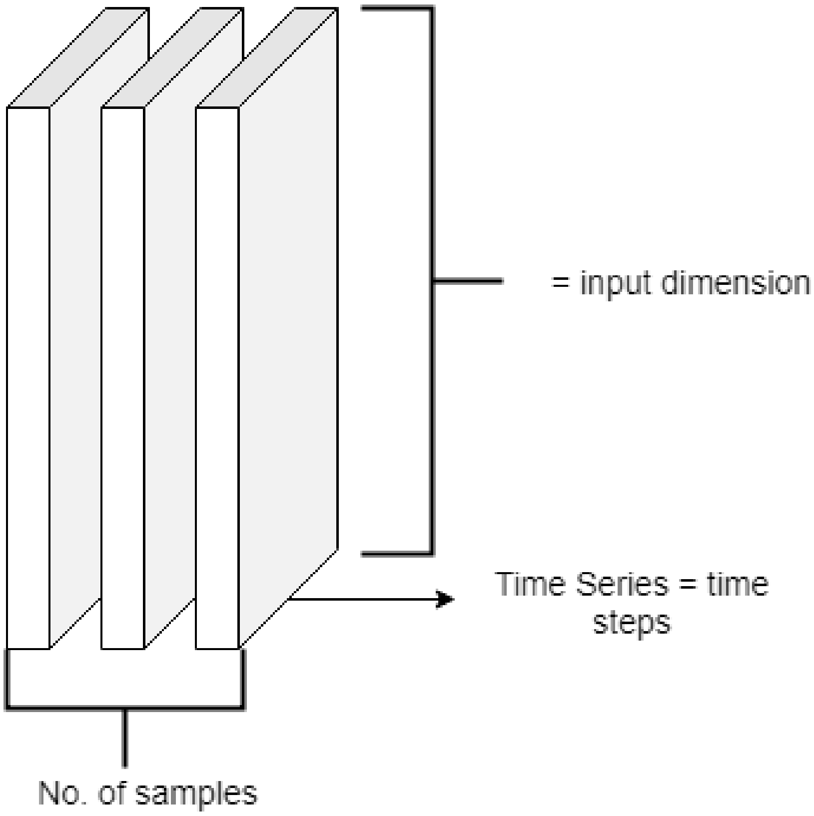

Figure 4.14: Input shape of LSTM

By using this dimension LSTM keeps track of the previous occurrence. So, inside our load data method we store our dataframe in matrix format. As index starts from 0 we increment our window size to 1. We declare an array and append data until maximum date with each index=index + window size. Then we reshape our array and split into train set and test set. We train our model with 90% of our data and test with the remaining 10%. Then we reshape our matrix into 3D matrix using

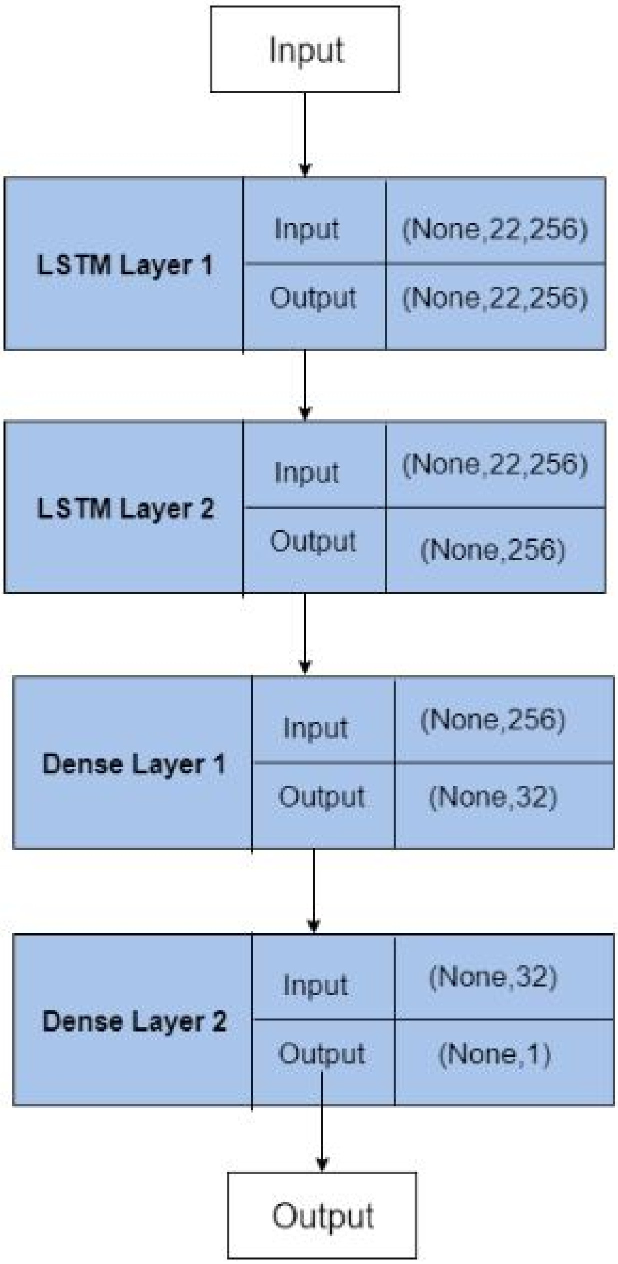

Figure 4.15: Long short term memory archiecture

Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(None, 22, 256)	272384
dropout_1 (Dropout)	(None, 22, 256)	0
lstm_2 (LSTM)	(None, 256)	525312
dropout_2 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 32)	8224
dense_2 (Dense)	(None, 1)	33
[l]Total params: 805,953		
Trainable params: 805,953		
Non-trainable params: 0		

Table 4.2: Model summary (LSTM)

np.reshape.
To train our model we used stacked LSTM. We know that LSTMs work on series of data, it means that the incorporation of a new layer adds a level of abstraction of input observations. It is imagined as a stack of layers. This is called Stacked LSTM. This architecture is described as an LSTM model of multiple LSTM layers. Here an upper LSTM layer produce a sequence output to the LSTM layer below. Specifically, one output per input time step, rather than one output time step for all input time steps. To create stacks of LSTM layers, we need a 3D array as input for the following layer. That means we require to change the configuration of the preceding LSTM layer to output a 3D array.
By changing the return sequences argument to True, we can get output after every node while the default value false means one output at last node. Our model is a Stacked LSTM with two hidden layers. We can also particularize the number of neurons in the layer. Their are three arguments to help us define number of neurons, initialize method and designate activation function. The first argument is for number of neurons, the second argument is for method initialization and last one is the activation argument. There is also option of manipulating network weights. Using uniform distribution we can generate small random numbers (between 0 and 0.5) and set them as network weights.
Next, we fit the model for only 300 epochs and a batch size of 511 for the training data set. We then compile the model with the loss function mean square error and adam optimizer. Because we want to show the loss and accuracy values. So, we specify metrics as accuracy of model where we can verify the shape of input and output shapes.

4.3 Experiment Results

4.3.1 Random Forest

The trained model is tested to make predictions on the test dataset with features only and then the result is compared with the known result. While performing regression, we have used Mean Square Error to compute the prediction error.

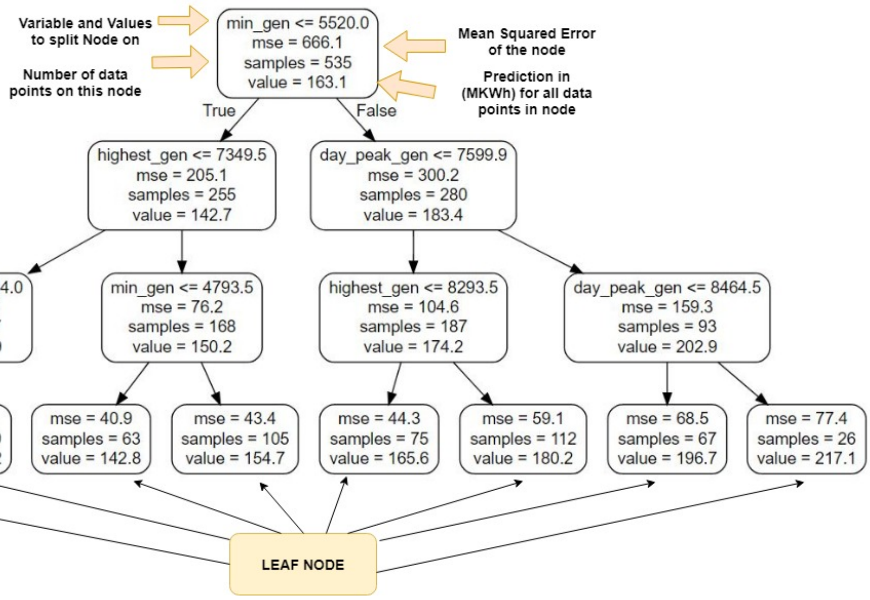

Figure 4.16: A random tree up to three levels

A random tree from the forest was examined as shown in Fig.4.16 with depth level of three for result analysis. By analyzing this tree, we can make prediction for new data points. While inspecting the tree, we can see that the root node has a 535 samples even though we have more than one thousand data points. This is due to bootstrap aggregation or bagging procedure discussed earlier, where we see that each tree in the forest is trained on a random subset of data generated on the basis of random sampling of the dataset with replacement.

Prediction Errors and Accuracy Score of RF	
Mean Squared Error (MSE)	0.15
Root Mean Squared Error (RMSE)	0.38
Accuracy(%)	98.24

Table 4.3: Prediction errors and accuracy score of Random Forest

As we go in depth in observing this particular tree, we see that only 3 variables are being used to make prediction and the rest of the features are not important in terms of making forecast. Maximum temperature, week of the day and other features are useless for forecasting. According to the tree, features like - the Minimum Generation (Generation end), Day-peak Generation (Generation end), and Highest Generation (Generation end) are most important for forecasting the load.

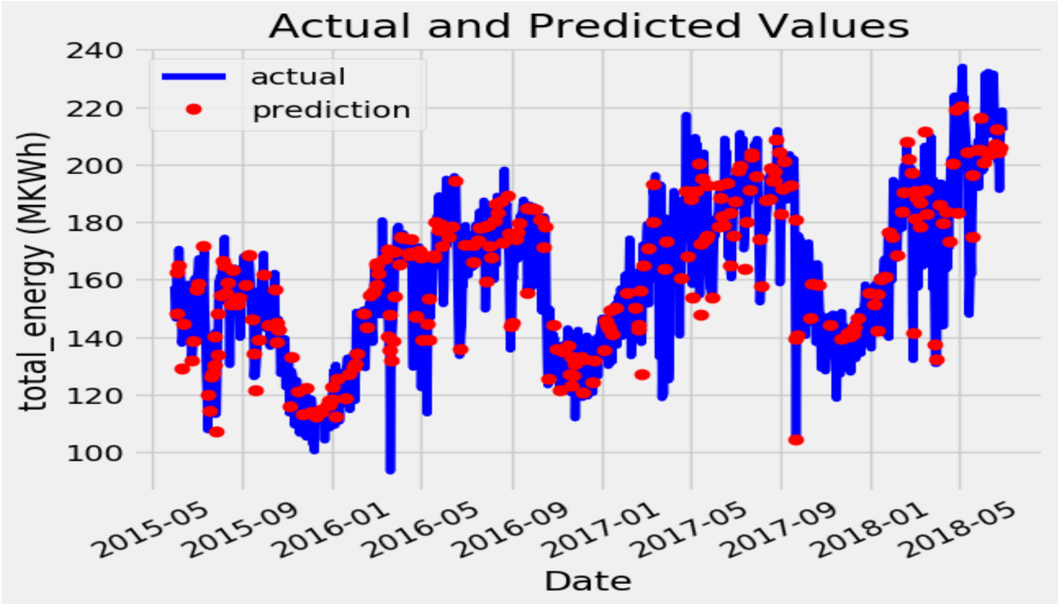

Figure 4.17: Actual vs. predicted electricity usage (RF)

The Fig.4.17 shows the load curve of actual vs. predicted load data and the table 4.3 shows the accuracy score along with the Mean Squared Error (MSE) Root Mean Squared Error (RMSE).

4.3.2 K Nearest Neighbor

The following step of kNN implementation is to actually make predictions using the test set we kept without the target column, We also check some commonly used error checking scores such as MSE and RMSE using the actual values.
By comparing the error scores with the previously applied algorithm we can see that the RMSE value is higher. This clearly indicates that kNN has performed poorly. There is not a huge difference in the RMSE value, but a plot for the predicted and actual values should provide a more clear understanding.

Prediction Errors and Accuracy Score of KNN	
Mean Squared Error (MSE)	0.18
Root Mean Squared Error (RMSE)	0.425
Accuracy(%)	98.11

Table 4.4: Prediction errors and accuracy score of KNN

After visualizing the comparison using plotting we can see in Fig.4.18 that although this learning algorithm is able to predict the load at many time points, it fails miserably at many points.

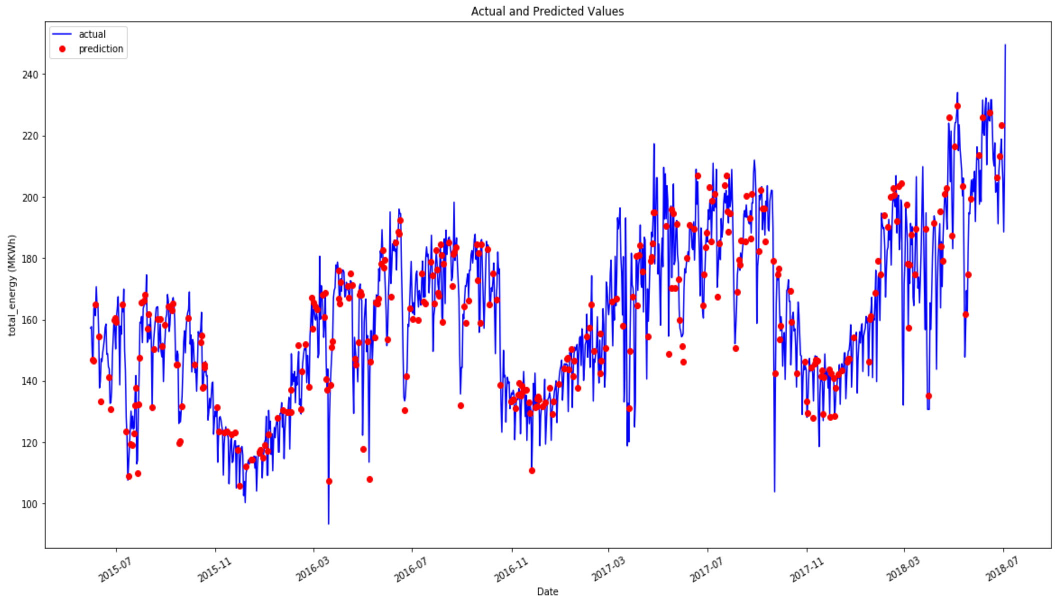

Figure 4.18: Actual vs. predicted electricity usage (KNN)

4.3.3 Long Short Term Memory

We have tested with LSTM network with up to 4 LSTM cell stacked top of one another. Stacking more than one LSTM cell made computation heavier but gave better results. We also improved our prediction by increasing the number of epochs. In the Fig.4.19 we can see the comparison between the actual energy consumption with the predicted power consumption of our model. We can see that predicted data points by a LSTM almost catches the pattern of electricity usages.

Prediction Errors of LSTM	
Mean Squared Error (MSE)	0.00917
Root Mean Squared Error (RMSE)	0.10

Table 4.5: Prediction errors of LSTM

It also generates lower RMSE and MSE score than random forest implementation as shown in table 4.5.
Looking at the loss plot in Fig.4.20 we can say our model has been trained and with the increasing amount of epochs the loss and val_loss both decreased to a convincing a point. We can also see our model converged at 300 epochs.

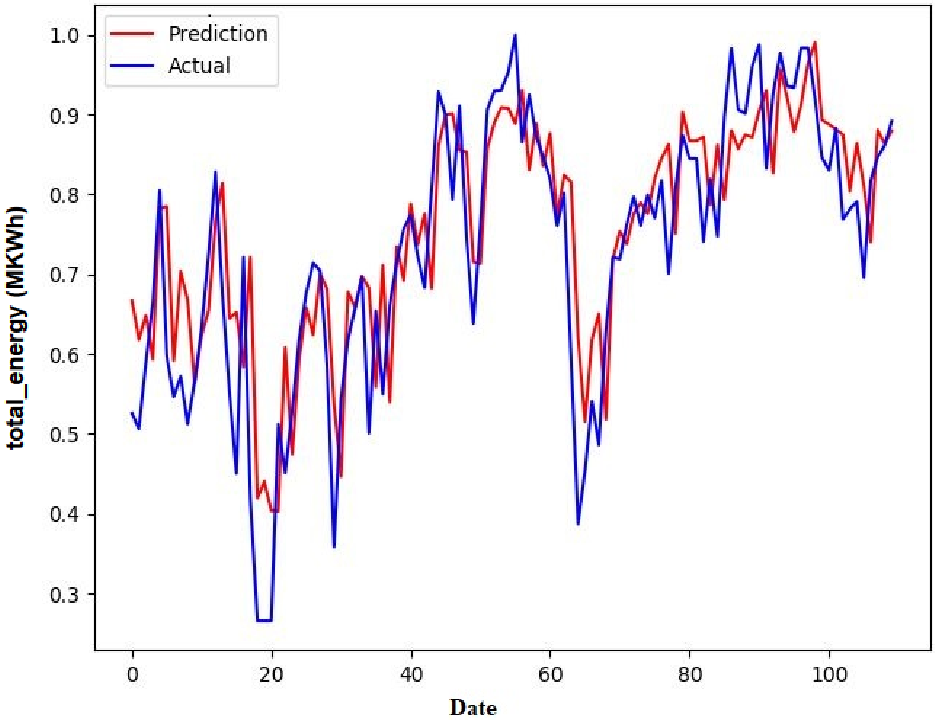

Figure 4.19: Actual vs. predicted electricity usage (LSTM)

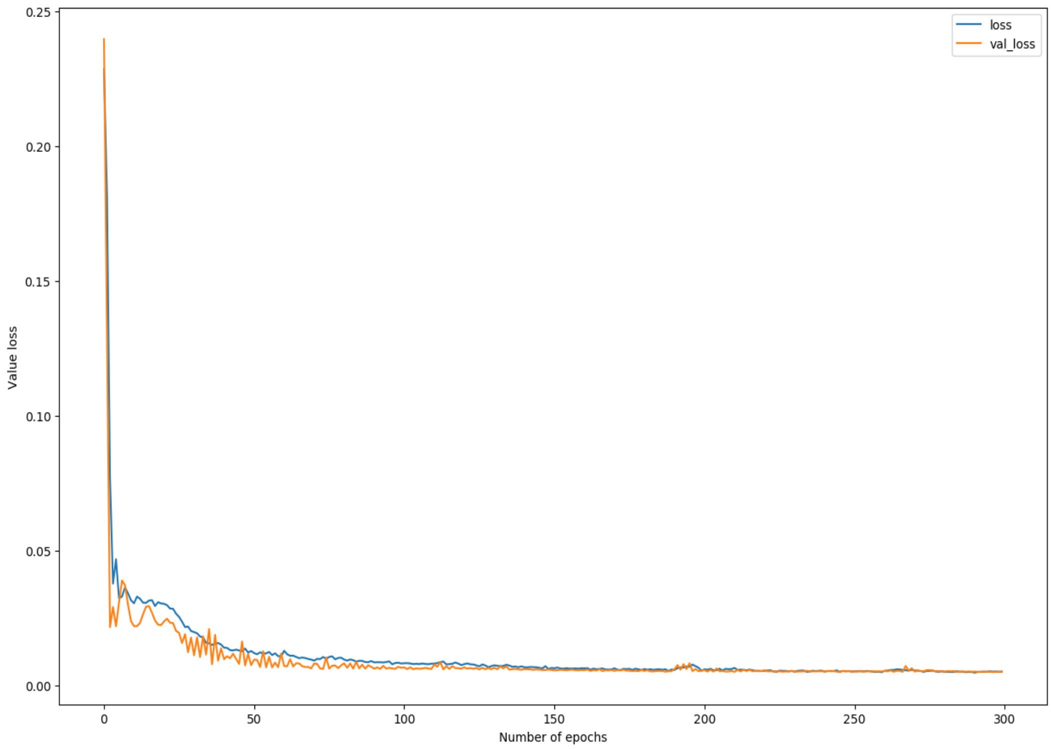

Figure 4.20: Convergence of error (LSTM)

4.4 Comparison and Result Summary

To compare the prediction result of the applied algorithm, we mainly focused on mean squared error (MSE) and Root Mean Square Error (RMSE). In the equation no 4.1 a mathematical equation that will give us the mean squared error for all our points is defined. In equation 4.2 the equation to calculate RMSE is shown where $(z_{fi} - z_{oi})^2$ is differences, squared and N is sample size.

$$MSE = 1/n \sum_{i=1}^{n} ((y_i - y_i')) \tag{4.1}$$

$$RMSE = \sqrt{\sum_{i=1}^{N} ((z_{fi} - z_{oi})^2 / N)} \tag{4.2}$$

As we can see LSTM generates a much less RMSE and MSE. As in recent days processor come with dedicated core and better performing GPU cost and computational power required for training a neural network is not an issue. Neural networks like LSTM has the ability to adopt input of variety of patterns and the ability of recognize those pattern.
According to our experiment LSTM has given better result with compared to Random Forest implementation. In the table 4.6 we can see all the error values generated in the applied algorithms.

2*Algorithms	Error	
	Root Mean Squared Error (RMSE)	Mean Squared Error (MSE)
Random Forest	0.38	0.15
K Nearest Neighbor	0.425	0.18
LSTM	0.10	0.00917

Table 4.6: Error comparison of applied algorithms

To visualize the result comparison better we plotted all the error values of the applied algorithms in a bar plot. In the figure 4.21 we can see that LSTM performs better than all the other regression analysis methods. Again among KNN and RF, RF provides a better result in comparison.

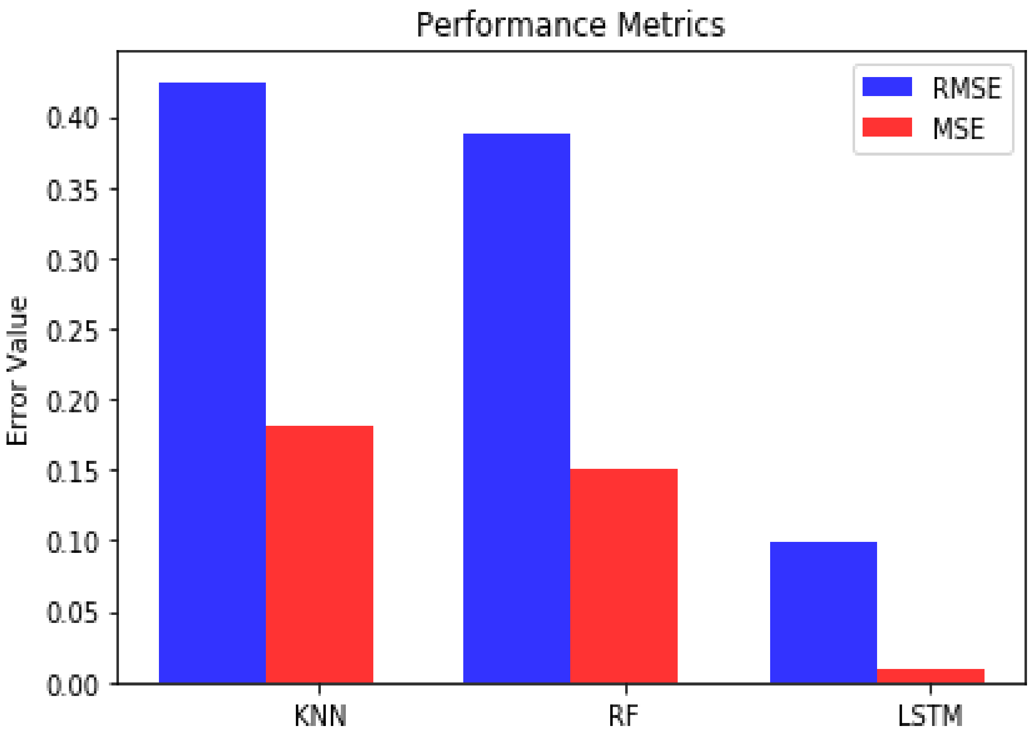

Figure 4.21: Error comparison of the models

Chapter 5

Conclusion and Future Work

5.1 Conclusion

With the fast growing economy of Bangladesh, energy shortage can become a severe impediment in the growth of our country. Though load shortage has been a buzz topic for a while, very few advanced researches took place referring to our energy generation conditions. Modern computation methodologies such as artificial intelligence can surely provide effective findings that can minimize the generation and demand gap.

In our research, we have successfully generated a compact dataset containing historical load data of three years (June, 2015 - June, 2018) from the data available as daily generation report (pdf format) in Bangladesh Power Development Boards website. Furthermore, we have tried to predict short term electricity demand with three trending machine learning algorithms. Different parameters that affect our prediction and how the outcome of our research can come handy to the power generation board and utility companies has also been discussed in this study. These results can guide electricity generation companies with the opportunity to do advance planning on maintenance and distribution of electric load. To prove the reliability of the system we have tested the system with real world data sets with proper error computation where LSTM model produced a better result than other two models.

5.2 Future Work

Sufficient energy supply is the precondition to ensure the striving economic growth of any country. The economy of Bangladesh is showing aspiring growth in all of its aspects and an effective power supply is a must to contribute to this growth. With this perspective in our minds, we plan to further work on our research in the near future. We target to develop a hybrid system that can identify important features recognized by the random forest model and then feed the data (filtered on the basis of impact factor of the features) to the LSTM model to see if we can forecast load more efficiently. To generate a database with more historical data that can be used in later stages of our future research is also included in our future plan.

Due to time restrictions we were not able to work with the zone wise load data available in BPDBs website. We plan to include the effect of zone wise load data to the ultimate load consumption along with predicting energy consumption zone wise so that we can improve the power supply according to demand in different areas. To add to this, considering different sources of energy production such as solar energy, biogas, microgrid which combined provide a huge amount of electricity to our national grid shall also be worked with to forecast the load effectively.

Bibliography

[1] E. B. Hunt, J. Marin, and P. J. Stone, "Experiments in induction", 1966.

[2] T. Dillon, K. Morsztyn, and K. Phua, "Short term load forecasting using adaptive pattern recognition and self-organizing techniques", in *Proc of the Fifth Power Sytem Computation Conference*, 1975, pp. 1–16.

[3] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and regression trees. belmont, ca: Wadsworth", *International Group*, p. 432, 1984.

[4] T. Gönen, "Load forecasting using box-jenkins methodology", *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*, vol. 3, no. 1, pp. 35–46, 1984.

[5] G. Gross and F. D. Galiana, "Short-term load forecasting", *In: Proceedings of the IEEE, pp*, vol. 1558, 1987.

[6] S. Rahman, "Formulation and analysis of a rule-based short-term load forecasting algorithm", *Proceedings of the IEEE*, vol. 78, no. 5, pp. 805–816, 1990.

[7] T. Dillon, S. Sestito, and S. Leung, "Short term load forecasting using an adaptive neural network", *International Journal of Electrical Power & Energy Systems*, vol. 13, no. 4, pp. 186–192, 1991.

[8] K. Y. Lee, Y. T. Cha, and J. H. Park, "Short-term load forecasting using an artificial neural network", *IEEE Transactions on Power Systems*, vol. 7, no. 1, pp. 124–132, Feb. 1992, ISSN: 0885-8950. DOI: 10.1109/59.141695.

[9] C. Cortes and V. Vapnik, "Support-vector networks", *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[10] T. K. Ho, "Random decision forests", in *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, ser. ICDAR '95, Washington, DC, USA: IEEE Computer Society, 1995, pp. 278–, ISBN: 0-8186-7128-9. [Online]. Available: http://dl.acm.org/citation.cfm?id=844379.844681.

[11] A. S. Weigend, M. Mangeas, and A. N. Srivastava, "Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting", *International Journal of Neural Systems*, vol. 6, no. 04, pp. 373–399, 1995.

[12] L. Breiman, "Bagging predictors", *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[13] A. S. Al-Fuhaid, M. A. El-Sayed, and M. S. Mahmoud, "Neuro-short-term load forecast of the power system in kuwait", *Applied Mathematical Modelling*, vol. 21, no. 4, pp. 215–219, 1997.

[14] C. J. Burges, “A tutorial on support vector machines for pattern recognition”, *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[15] S. R. Gunn *et al.*, “Support vector machines for classification and regression”, *ISIS technical report*, vol. 14, no. 1, pp. 5–16, 1998.

[16] L. Breiman, “Random forests”, *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[17] H. S. Hippert, C. E. Pedreira, and R. C. Souza, “Neural networks for short-term load forecasting: A review and evaluation”, *IEEE Transactions on power systems*, vol. 16, no. 1, pp. 44–55, 2001.

[18] S. Tzafestas and E. Tzafestas, “Computational intelligence techniques for shortterm electric load forecasting”, *Journal of Intelligent and Robotic Systems*, vol. 31, p. 7, 2001.

[19] A. T. Lora, J. C. Riquelme, J. L. M. Ramos, J. M. R. Santos, and A. G. Expósito, “Influence of knn-based load forecasting errors on optimal energy production”, in *Portuguese Conference on Artificial Intelligence*, Springer, 2003, pp. 189–203.

[20] B.-J. Chen, M.-W. Chang, *et al.*, “Load forecasting using support vector machines: A study on eunite competition 2001”, *IEEE transactions on power systems*, vol. 19, no. 4, pp. 1821–1830, 2004.

[21] F. Erchiqui and N. Kandil, “Neuronal networks approach for characterization of softened polymers”, *Journal of reinforced plastics and composites*, vol. 25, no. 5, pp. 463–473, 2006.

[22] S. Fan and L. Chen, “Short-term load forecasting based on an adaptive hybrid method”, *IEEE Transactions on Power Systems*, vol. 21, no. 1, pp. 392–401, Feb. 2006, ISSN: 0885-8950. DOI: 10.1109/TPWRS.2005.860944.

[23] N. Kandil, R. Wamkeue, M. Saad, and S. Georges, “An efficient approach for shorterm load forecasting using artificial neural networks”, in *2006 IEEE International Symposium on Industrial Electronics*, IEEE, vol. 3, 2006, pp. 1928–1932.

[24] J. Pereira, R. Salgado, T. Ohishi, and R. Ballini, “Short-term load forecasting using support vector machines”, in *IEEE PES Transmission and Distribution Conference*, 2006.

[25] J. Yang, “Power system short-term load”, PhD thesis, Technische Universität, 2006.

[26] G. K. Tso and K. K. Yau, “Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks”, *Energy*, vol. 32, no. 9, pp. 1761–1768, 2007.

[27] R. Weron, *Modeling and forecasting electricity loads and prices: A statistical approach*. John Wiley & Sons, 2007, vol. 403.

[28] M. Jamaluddin, “Draft saarc regional trade study, country report—Bangladesh”, 2008.

[29] H. Hahn, S. Meyer-Nieberg, and S. Pickl, “Electric load forecasting methods: Tools for decision making”, *European Journal of Operational Research*, vol. 199, no. 3, pp. 902–907, Dec. 2009. DOI: 10.1016/j.ejor.2009.01.062. [Online]. Available: https://doi.org/10.1016/j.ejor.2009.01.062.

[30] M. A. H. Mondal, W. Boie, and M. Denich, “Future demand scenarios of Bangladesh power sector”, *Energy Policy*, vol. 38, pp. 7416–7426, Nov. 2010. DOI: 10.1016/j.enpol.2010.08.017.

[31] Q. Mu, Y. Wu, X. Pan, L. Huang, and X. Li, “Short-term load forecasting using improved similar days method”, in *2010 Asia-Pacific Power and Energy Engineering Conference*, Mar. 2010, pp. 1–4. DOI: 10.1109/APPEEC.2010.5448655.

[32] S. A.-h. Soliman and A. M. Al-Kandari, *Electrical load forecasting: modeling and model construction*. Elsevier, 2010.

[33] Y. Xu, Z. Y. Dong, J. Zhao, K. Meng, and K.-P. Wong, “Transient stability assessment based on data-structure analysis of operating point space”, in *IEEE PES General Meeting*, IEEE, 2010, pp. 1–7.

[34] B. U. Islam, “Comparison of conventional and modern load forecasting techniques based on artificial intelligence and expert systems”, *International Journal of Computer Science Issues (IJCSI)*, vol. 8, no. 5, p. 504, 2011.

[35] Y.-Y. Cheng, P. P. Chan, and Z.-W. Qiu, “Random forest based ensemble system for short term load forecasting”, in *2012 International Conference on Machine Learning and Cybernetics*, IEEE, vol. 1, 2012, pp. 52–56.

[36] J. W. Taylor and R. D. Snyder, “Forecasting intraday time series with multiple seasonal cycles using parsimonious seasonal exponential smoothing”, *Omega*, vol. 40, no. 6, pp. 748–757, 2012.

[37] G. M. Khan, A. R. Khattak, F. Zafari, and S. A. Mahmud, “Electrical load forecasting using fast learning recurrent neural networks”, in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2013, pp. 1–6.

[38] J.-M. Cha and B.-H. Ku, “A study on the summer and winter load forecasting by using the characteristics of temperature changes in korean power system”, *Journal of International Council on Electrical Engineering*, vol. 4, no. 4, pp. 293–296, 2014. DOI: 10.1080/22348972.2014.11011885. eprint: https://doi.org/10.1080/22348972.2014.11011885. [Online]. Available: https://doi.org/10.1080/22348972.2014.11011885.

[39] G. Dudek, “Short-term load forecasting using random forests”, in *Intelligent Systems' 2014*, Springer, 2015, pp. 821–828.

[40] M. Ishrat Malek, M. Hossain, and M. A. R. Sarkar, “Production and utilization of natural gas in bangladesh”, Jan. 2015.

[41] C. Lier, “Applying machine learning techniques to short term load forecasting”, 2015.

[42] Sustainable and R. E. D. Authority, “Energy efficiency and conservation master plan up to 2030”, in, Sustainable et al., 2015.

[43] E. E. Community, *Electric load forecasting: Advantages and challenges*, http://engineering.electrical-equipment.org/electrical-distribution/electric-load-forecasting-advantages-challenges.html, 2016. [Online]. Available: http://engineering.electrical-equipment.org/electrical-distribution/electric-load-forecasting-advantages-challenges.html.

[44] N. Huang, G. Lu, and D. Xu, "A permutation importance-based feature selection method for short-term electricity load forecasting using random forest", *Energies*, vol. 9, no. 10, p. 767, 2016.

[45] Meng, Xiao Fang, Zhang, Shu Xuan, Wang, Jun, and Wu, Shi Hong, "Load forecasting method considering temperature effect for distribution network", *MATEC Web Conf.*, vol. 63, p. 05 028, 2016. DOI: 10.1051/matecconf/20166305028. [Online]. Available: https://doi.org/10.1051/matecconf/20166305028.

[46] V. Rodriguez-Galiano, M. Sánchez Castillo, J. Dash, P. Atkinson, and J. Ojeda-Zujar, "Modelling interannual variation in the spring and autumn land surface phenology of the european forest", *Biogeosciences*, vol. 13, pp. 3305–3317, Jun. 2016. DOI: 10.5194/bg-13-3305-2016.

[47] O. Valgaev, F. Kupzog, and H. Schmeck, "Low-voltage power demand forecasting using k-nearest neighbors approach", in *2016 IEEE Innovative Smart Grid Technologies-Asia (ISGT-Asia)*, IEEE, 2016, pp. 1019–1024.

[48] R. Zhang, Y. Xu, Z. Y. Dong, W. Kong, and K. P. Wong, "A composite k-nearest neighbor model for day-ahead load forecasting with limited temperature forecasts", in *2016 IEEE Power and Energy Society General Meeting (PESGM)*, IEEE, 2016, pp. 1–5.

[49] N. Ammar, M. Sulaiman, A. Fateh, and A. F. Mohamad Nor, "Analysis load forecasting of power system using fuzzy logic and artificial neural network", *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 9, pp. 181–192, Sep. 2017.

[50] K. Anoop and K. Kanchana, "Short term load forecasting using fuzzy logic control", in *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, IEEE, 2017, pp. 2983–2986.

[51] Y. Chen, P. Xu, Y. Chu, W. Li, Y. Wu, L. Ni, Y. Bao, and K. Wang, "Short-term electrical load forecasting using the support vector regression (svr) model to calculate the demand response baseline for office buildings", *Applied Energy*, vol. 195, pp. 659–670, 2017.

[52] P. Ganguly, A. Kalam, and A. Zayegh, "Short term load forecasting using fuzzy logic", May 2017.

[53] T. Li, B. Wang, M. Zhou, and J. Watada, "Short-term load forecasting using optimized lstm networks based on emd", *arXiv preprint arXiv:1809.10108*, 2018.

[54] *Bangladesh power development board*, http://www.bpdb.gov.bd/bpdb_new/index.php/site/daily_generation, (Accessed on 04/29/2019), 2019.

[55] *Electricity sector in bangladesh - wikipedia*, https://en.wikipedia.org/wiki/Electricity_sector_in_Bangladesh, (Accessed on 05/14/2019),

[56] *Power system master plan 2016 summary*, https://powerdivision.portal.gov.bd/sites/default/files/files/powerdivision.portal.gov.bd/page/4f81bf4d_1180_4c53_b27c_8fa0eb11e2c1/(E)_FR_PSMP2016_Summary_revised.pdf, (Accessed on 05/14/2019),