SHORT-TERM RESIDENTIAL LOAD FORECASTING USING DEEP LEARNING TECHNIQUES

A PROJECT REPORT

submitted by

SANJAY STEEPHEN
(Reg. No. TKM20EEII16)

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Technology
in
Electrical and Electronics Engineering

with specialisation in

Industrial Instrumentation and Control

Department of Electrical and Electronics Engineering
TKM College of Engineering
Kollam - 691005
KERALA

JULY 2022

DECLARATION

I undersigned hereby declare that the project report entitled "**Short-Term Residential Load Forecasting Using Deep Learning Techniques** ", submitted for partial fulfillment of the requirements for the award of degree of Master of Technology in Electrical and Electronics Engineering with specialisation in Industrial Instrumentation and Control, of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of *Dr. Sheeba R*, Professor, Department of Electrical and Electronics Engineering. This submission represents my ideas in my own words and where ideas or words of others have been included. I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Kollam
July 01, 2022

SANJAY STEEPHEN

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

TKM COLLEGE OF ENGINEERING

KOLLAM - 691005

CERTIFICATE

This is to certify that the report entitled " **Short-Term Residential Load Forecasting Using Deep Learning Techniques** " submitted by **SANJAY STEEPHEN** , **(Reg. No. TKM20EEII16)** of fourth semester to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Technology in Electrical and Electronics Engineering with specialisation in Industrial Instrumentation and Control, is a bonafide record of the project work done by him under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Dr. Sheeba R
Project Supervisor
Professor
Department of EEE
TKM College of Engineering

Prof. Sumayya Jaleel
Project Coordinator
Assistant Professor
Department of EEE
TKM College of Engineering

Prof. Shanavas T N
PG Coordinator
Associate Professor
Department of EEE
TKM College of Engineering

Dr. Sabeena Beevi K
Head of the Department
Associate Professor
Department of EEE
TKM College of Engineering

Acknowledgement

A lot of effort and hard work has been put into this project in course of its presentation. However, it would not have been possible without the kind support and help of many individuals and other sources. I would like to extend my sincere thanks to all of them. I take this opportunity to express my deep sense of gratitude and sincere thanks to all who helped me to complete this project report successfully.

I express my sincere thanks to *Dr. T A Shahul Hameed*, Principal, TKM College of Engineering, for his encouragement in the completion of my project.

I thank *Dr. Sabeena Beevi K*, Associate Professor, Head of the Department, Department of Electrical and Electronics Engineering, *Dr. Imthias Ahamed T P*, Professor, Department of Electrical and Electronics Engineering and *Prof. Shanavas T N*, Associate Professor, PG Co-ordinator, Department of Electrical and Electronics Engineering, for their support and cooperation.

I am greatly thankful to *Dr.Sheeba R*, Professor, Project Guide, Department of Electrical and Electronics Engineering and *Mr. Naufal N*, for their supervision, assistance and helpful suggestions.

I am deeply indebted to *Prof. Sumayya Jaleel*, Assistant Professor, Project Coordinator, Department of Electrical and Electronics Engineering, *Prof. Amal A*, Assistant Professor, Department of Electrical and Electronics Engineering and *Prof. Sumod Sundar*, Assistant Professor, Centre for Artificial Intelligence, for their excellent guidance, positive criticism and valuable comments.

Finally I thank my parents, friends near and dear ones who directly and indirectly contributed to the successful completion of my project project.

SANJAY STEEPHEN

Abstract

Smart energy management systems have become more popular as the consumption of energy is increasing rapidly. So, in order to monitor the daily energy consumptions in real time smart meters are used. Load forecasting is very important for power system management as it helps in maximum utilization of power generation plants, reliable and efficient operation of the system. In smart homes, the smart meter data are used to forecast the load and it can be even used for a neighborhood. F orecasting of electrical loads can be done using different deep learning techniques and can be used for demand management. Different methods employed includes Long Short-Term Memory (LSTM), Bi-directional Long Short-Term Memory (Bi-LSTM) and Gated Recurrent Unit (GRU). The proposed method for forecasting of energy consumption consists of data preprocessing, model generation and validation. Performance of the models are validated using evaluation metrics like R-squared ($\mathbf{R}^2$), Mean Absolute Error (MAE), Mean Square Error (MSE) and Root Mean Square Error (RMSE). The error metrics are then compared to find out the accurate model. The main advantage of load forecasting is that we can reduce the energy wastage and increase the efficiency of energy usage.

Contents

List of Tables

List of Figures

Abbreviations

$\mathbf{Bi-LSTM}$	Bi-directional Long Short-term Memory
DNN	Deep Neural Networks
GRU	Gated Recurrent Unit
LSTM	Long Short-term Memory
MAE	Mean Absolute Error
MSE	Mean Square Error
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network

Notations

N	Total number of values in the test set
y_i	Actual value
$\hat{y}_i$	Predicted value

Chapter 1

INTRODUCTION

1.1 Overview

With the rise in population and an overall increase of the energy structure, electricity demand is fast-growing. So, smart grids are used to meet the energy demand. The key attribute of demand-side managing in smart grids is load forecasting, as it helps to make efficient and effective choices [1]. Load forecasting is an intricate multi-variable and multi-dimensional approximation problem where forecasting methods such as curve fitting using numerical methods do not provide accurate results as the random trends are not considered accurately and that's why machine learning algorithms are used [2].

Nowadays, artificial intelligence-based approaches that are formed upon data engineering and machine learning techniques are so accepted and broadly used in different areas. But for a large dataset deep learning methods gives the prefect approach. Deep learning techniques consist of multiple layers which uses data abstraction for feature extraction. The new form of deep Recurrent Neural Networks (R-NNs) with more precise and more stable performance, namely Long Short-Term Memory (LSTM) network, has been used in different works such as short-term load demand and power fluctuations forecasting tasks [3]. In order to develop a machine learning model for load forecasting it is important to understand the different parameters on which electricity demand is dependent. The different parameters on which it depends are time, historical trends, weather, humidity, electricity price, etc. In LSTM models used for load forecasting, adding layers and hidden nodes in in excess can cause over fitting. It works remarkably well on training dataset but flops when it enters a real-world situation [4].

In order to capture the appliance level behavior in low resolution data Hidden Markov Model (HMM) is used. HMM is used since it does the individual modelling of appliances and improve the results. It works well for single devices but not so effective in case of complex appliances. HMM is very time consuming due to the individual modelling in detail, limits its practical use [5]. The computational difficulty rises with the number of devices. So, to overcome the disadvantages deep neural networks (DNN) are used. Regardless of the number of appliances or the electricity consumption from non-target appliances the deep learning models identify the activities and estimates the consumption of the appliances regardless of their number [6].

Energy consumption pattern at a household can be of three types. The regular consumption pattern which can be identified from analyzing the historical data, uncertain consumption pattern due to weather conditions and noise components [7]. Due to the uncertainty and unpredictability of household energy consumption, deep learning techniques like LSTM, Bi-LSTM, GRU etc. are used. LSTM is a more advanced type of RNN which can be used for time series data. Bi-LSTM is different from LSTM as it processes data in forward and backward directions. GRU is a LSTM with only two gates. The proposed deep learning models are then evaluated based on accuracy to find the appropriate method.The main contributions of this work are, a deep learning model is built using LSTM, Bi-LSTM and GRU and quantitative analysis are performed through evaluation metrics.

1.2 Objective

New methods are being developed in order to conserve the energy usage and are being implemented to overcome the existing methods. Smart energy meters are being developed on the idea of increasing the energy efficiency, real time monitoring of energy usage and for the demand side management. The smart meter readings are then used for load forecasting, which helps to reduce the cost of production of power utility companies, give feedback to customers, helps in demand forecasting and to meet future commercial and environmental impacts. Advanced deep learning techniques like Long Short-Term Memory (LSTM), Bi-directional Long Short-Term Memory (Bi-LSTM) and Gated Recurrent Unit (GRU) are used for load forecasting and their performance are evaluated using different error metrics.

1.3 Organization of the Thesis

The entire thesis is organized as follows. It consists of five chapters. Chapter 1 is a brief introduction of the thesis, objectives and scope about the same. Chapter 2 deals with literature review about load forecasting methods, datasets and different techniques used. Chapter 3 describes the methodology followed. Chapter 4 discuss about the result and analysis for the forecasting models. Finally, Chapter 5 gives the conclusion and the future scope.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

Load forecasting is very important in power system management for the efficient utilization of the system. The evaluation of load in advance is generally known as load forecasting. It is necessary for power system planning, for demand side management and peak load reduction. The peak load reduction can be used to increase the profits and for reducing the energy cost. If the loads are known in advance the utility companies can manage their operation with better planning.

2.2 Load forecasting

It is the process used for forecasting future electric load, given previous trends and weather information. Various type of load forecasting based on the range of forecast are Short term load forecasting, Medium term load forecasting and Long term load forecasting [8].

2.2.1 Short term load forecasting

The period ranges from one hour to one week. It helps us to understand the load consumption and can be used to avoid overloading. It also gives an idea about daily operations which can be used for system management [9], [10].

2.2.2 Medium term load forecasting

The operating period ranges from one week to one year. The forecasts for different time horizons are important for different operations within a utility company. It is used for the purpose of scheduling fuel supplies and unit management.

2.2.3 Long term load forecasting

The operating period is longer than a year. It is used to provide electric utility company management with accurate forecast of future needs for expansion, equipment purchases or staff hiring.

2.3 Advantages and Disadvantages of Load Forecasting

2.3.1 Advantages

1. Utility companies can plan their operations well since the future consumption is known, so as to meet the demand.

2. Resources necessary to operate the generating plants can be reduced without affecting their operations.

3. It helps to plan the size, location and type of the future generating plants according to the needs and ensure their maximum utilization.

2.3.2 Disadvantages

1. The accuracy of the forecast may vary, as it depends on different parameters and on how the data is interpreted.

2. Decision made on inaccurate forecast can cause financial losses.

For different operations, the forecast for different time zones is used within a power utility company. All these forecasts are different as well. The different techniques used are statistical techniques or artificial intelligence algorithms such as regression, neural networks, fuzzy logic and expert systems.

The smart grid operations use smart meters for automatic meter reading and load forecasting. Smart meter is a device that records electricity usage and permits bi-directional communication between the utility company and the resident for monitoring and billing [11], [12]. It provides the customer with a better idea about the energy consumed and the price. The utility companies generates electricity and provide to the consumers. The meter records the energy in a periodic time like hourly or in minutes. The smart meter output is time series data. The most important factors affecting the electricity consumption pattern of the resident are environmental factors and socioeconomic factors. The environmental factors include weather, temperature, and humidity of the resident's locality.

The changes in the climatic conditions and the lifestyle of the residents affect the electricity consumption patterns, making the load forecasting more difficult. Smart meters used for monitoring residents energy usage gives an idea about the load consumption pattern and socio-economic nature of the resident. It also gives idea about the appliances used, usage time and changes of the load with external factors like climatic conditions. The load patters changes accordingly to seasons and are needed to be addressed. The different factors affecting the load are demographic factors, time factors, weather factors, meteorological factors and pricing factors [13]. The meteorological factors were analysed to find out their correlation with load and the related features are given into the forecasting model as shown in Figure 2.1 to improve the prediction accuracy. Correlation metrics are used to find out the correlation coefficients between variables, so that the best features can be found out for load forecasting. A high degree of correlation between variables makes them unreliable and are excluded. External factors like temperature and humidity if considered makes load forecasting better and reduces the computational errors [14].

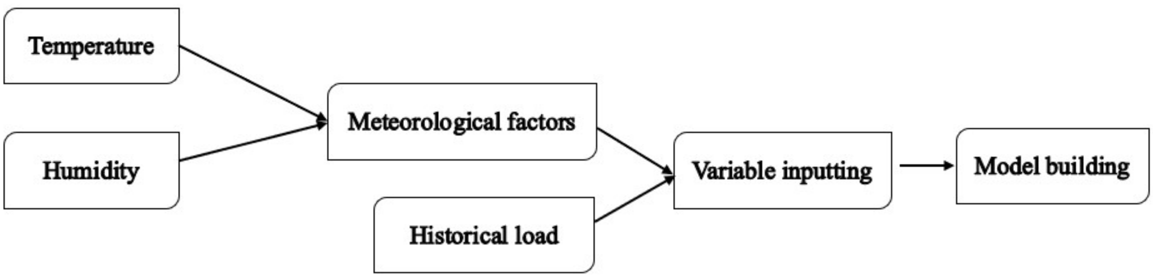

Figure 2.1: Load forecasting model

2.4 Data acquisition

2.4.1 REFIT

The dataset consists of 20 buildings in England (Loughborough area) from the years 2013 to 2015. The information comprises of active power tested at each 8 seconds for both mains and discrete devices. It contains the most data among the other datasets and are used in creating deep learning models. We anticipated that the huge quantity of data analyses would be able to generalize the trained models, that are used to further unobserved households. Beforehand training the models, the complete information is reviewed.

2.4.2 UK-DALE

UK-DALE (United Kingdom Domestic Appliance Level Dataset) is used for performance evaluation of deep neural networks [15]. The dataset contains both aggregated and disaggregated real power measurements along with selected submeter appliances from five households. The datasets were sampled at a 6 second interval for applying data augmentation algorithm. The power requirements of all appliance were represented by the aggregated data and was recorded using a smart energy meter.

Smart plugs are used in individual appliances to record their power demands and to measure the disaggregated power data. Each target appliances have labelled data in the UK-DALE set. So, in order to improve the size of training data appliance profile where offset, augmented and duplicated.UK- DALE has a lot of misplaced data and the gaps can be large. The gaps are considered as zeroes for easy estimation of consumption of appliances. The performance evaluation is affected due to the gaps and we separate the data at gaps into segments. For testing and training we use the longest and second longest segments.

2.4.3 REDD

The Reference Energy Disaggregation Data Set (REDD), is an easily accessible database comprising of comprehensive power utility of 6 buildings in US, issued with the objective of encouraging additional study on the field of energy disaggregation. It contains different frequency meter readings for small time interval (amongst a few weeks and a few months). Measurements comprise of readings with 1sec sampling period and numerous devices with 3sec sampling period. Also, high-frequency current and voltage readings are offered at a sample frequency of 15KHz and the span of the readings are among 3 and 19 days.

2.4.4 Smart home dataset

A real-world dataset containing the energy consumption values of different appliances of a household are used. The dataset was downloaded from Kaggle and the dataset used for the model creation consists of various parameters that affected the electricity consumption like temperature, pressure, humidity, wind speed, dew point, visibility etc. along with the appliance's consumption data. The data consist of the measurements with an interval of 1 minute for 350 days of a household. The dataset also contains the local weather information's and the data are represented in KW.

2.5 Deep learning techniques

For load forecasting, different techniques are used for getting better results. The main methods are divided into two, statistical methods and machine learning methods. Statistical methods such as the Auto regressive Integrated Moving Average model (ARIMA) and exponential smoothing method are simple and does not give the complex correlation between the different features of the load dataset. Machine learning techniques like Support Vector Machine (SVM) and decision tree give a better performance and are widely used in load forecasting. Most recently deep learning techniques like Convolutional Neural Network (CNN) and Recurrent Neural Networks (RNN) are of great importance as they give better prediction results. Advanced form of RNN like LSTM, Bi-LSTM and GRU are also used [16], [17], [18], [19].

2.5.1 Sequence-to-Sequence

It operates by converting a sequence into another, which consists of an encoder RNN, which recognize the input sequence along with a decoder RNN that decodes the assumed vector thus creating an output [20]. Encoder network changes the input to a vector that gives its output into the decoder network such that it forms a new series. During each stage of repetition in the encoder network, a new word is given as input which is used in the succeeding stage by the following stage [21]. When the decoder collects the last state from the encoder, it uses a distinct probability distribution (on the input at every stage) for predicting the output considering a loss function. The prototype used attempts to plot the data of the device which is monitored by creating a regressive map among them [22].

2.5.2 Multi task learning

Multi task learning (MTL) is a machine learning algorithm which involves multiple related tasks optimized by multiple loss functions [23]. This contributes to performance improvement as multiple related tasks are learned together. It involves sharing knowledge to learn common features between different tasks. MTL consist of one input layer followed by multiple shared layers to avoid overfitting and two output layers. The first output layer gives the energy disaggregation and the other gives the status detection. The energy disaggregation layer provides the appliance level energy consumption, load pattern etc. But for an effective energy management system (EMS) the operation time, frequency of use must also be known. So, in order to achieve accurate results, the load status detection layer is used. During off periods the energy disaggregation output produce noisy outputs. If the device is not active the status detection gives zero and the energy disruption results can be improved by taking the product of two outputs, which eliminates the noisy predictions.

2.6 Conclusion

This chapter addressed the literature review about load forecasting, the different types of load forecasting and factors affecting the forecast. Different datasets along with different deep learning techniques were also explained in detail.

Chapter 3

METHODOLOGY

3.1 Introduction

Deep learning-based load forecasting models are built on three main steps, namely data preprocessing, model training and model validation. The dataset contains different parameters that affects the forecast and are normalized to ensure consistency. Correlation between different factors is found out to find out the best features influencing the load. The different deep learning models are trained with the dataset and are evaluated using the test set , which are not seen in the training period.

3.2 Block diagram of the proposed method

The block diagram representation of the proposed method is represented in Figure 3.1. The main elements of the block diagram are dataset, data preprocessing , load forecasting, assessment of prediction model and comparison of prediction models.The dataset used to the model creation was from a smart home having a smart meter for load monitoring, were the energy consumption of the resident are monitored and the data are being communicated with both the user and the utility company. The dataset contains readings of 350 days over a time span of one minute.

The dataset are preprocessed to avoid inaccurate data and to avoid the uncertainty in prediction. The most significant features are found out by using correlation matrix and the uncorrelated features are avoided. Normalisation are done on the data to make it into a common scale.

The proposed model created was a comparison between different deep learning methods for load forecasting. The main methods used in this work are LSTM, Bi-LSTM and GRU. LSTM is a type of RNN that are capable of learning long term dependencies and has a chain like structure. It consists of an input gate, output gate and a forget gate. Unlike LSTM, the Bi-LSTM adds one more layer so that the information flows in both directions i.e.: in forward and backward directions. GRU is an advanced form of RNN and is similar to LSTM. The main difference is that it consists of only two gates, the reset gate and update gate.

The evaluation of different deep learning models are then evaluated based on performance metrics like R-squared, Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Square Error (RMSE). Finally the models are compared based on the performance metrics values to find out the most appropriate model for load forecasting.

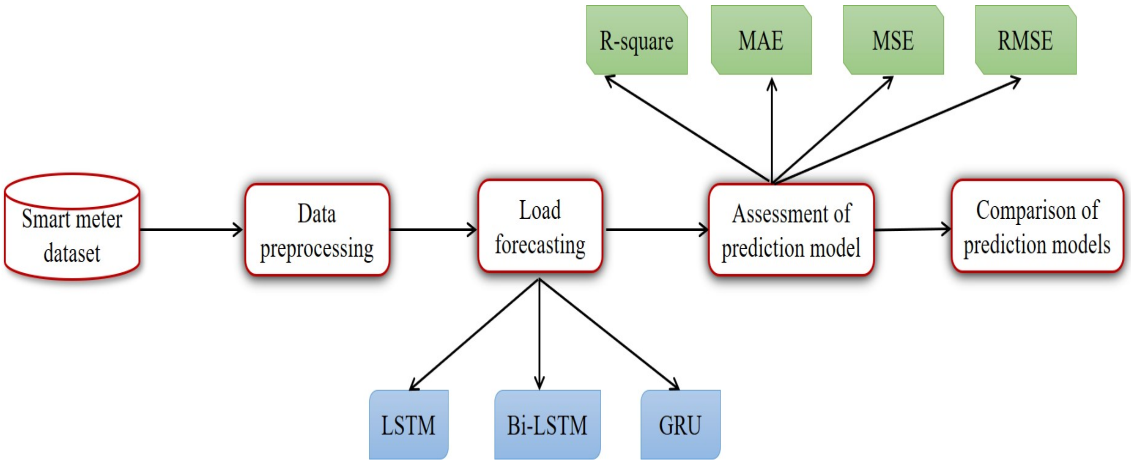

Figure 3.1: Block diagram

3.3 Data processing

Datasets used for model creation determines the performance of learning model, hence large datasets are used. Quality datasets are necessary to increase the correctness of the model created. Data pre-processing is done carefully with real world situations as datasets and the only problem is that it takes lot of time to collect the diverse datasets and it's not feasible to create learning models for diverse appliances.

3.3.1 Correlation matrix

It shows the relation between different variables arranged in rows and columns as a matrix. The matrix gives idea about the correlation coefficient of different variables. The values of the correlation matrix range from -1 to1. Feature selection is an important task in machine learning, so we use correlation matrix for feature selection by finding out the relationship or similarity among different variables. Feature selection also helps us to reduce the computational time, improve model accuracy and provides a better understanding of the data. The uncertainty of the predicted results can be avoided by using related data, so stronger the correlation greater will be the prediction accuracy.

3.3.2 Feature selection

Datasets are analysed to find out the variation in the features and the best are used to the model creation. Data pre-processing techniques like correlation metrics are important, as they improve the prediction accuracy by using the most related data. The load curve may change based on different factors like meteorological data, demographic data etc. A residents usage can change depending on time, number of people, income, number of rooms, appliances and according to climate. Drastic changes are observed during climatic changes, seasonal variations, week-days, temperature and humidity changes etc. The accuracy of the predicted data determines the smart grid performances and are of great importance for meeting economic and environmental objectives.

The dataset used for the model generation was in CSV format, so it was converted into an excel sheet for easy understanding of the data and the different parameters included. Also the excel sheet was used to generate graphs regarding any parameters by simply plotting the data. The graphs plotted in the excel was used to analyze the dataset. After converting the CSV file to an excel sheet the maximum load curves of different appliances were plotted by creating a new sheet from the excel sheet, which contain the peak load values of the corresponding appliances. Each appliances had a unique load curve which indicates its load usage and the peak load values. The dataset used is very important for the model creation as the accuracy of the data determines the model effectiveness and model must also be changed as the dataset changes.

3.3.3 Load curve generation

The dataset consists of data from different appliances in a particular household along with the data of different parameters which may affect the energy consumption. So, in order to find out the energy use pattern of different appliances the load curves of various appliances were generated and the plot of the appliances were studied to find out the variations. The maximum load curves of the appliances were created by using a separate excel sheet apart from the one containing the original dataset. The new sheet composed of values from the maximum load curves of different appliances, which was calculated from the graphs generated. The total power consumption vs time graph is represented in Figure 3.2.

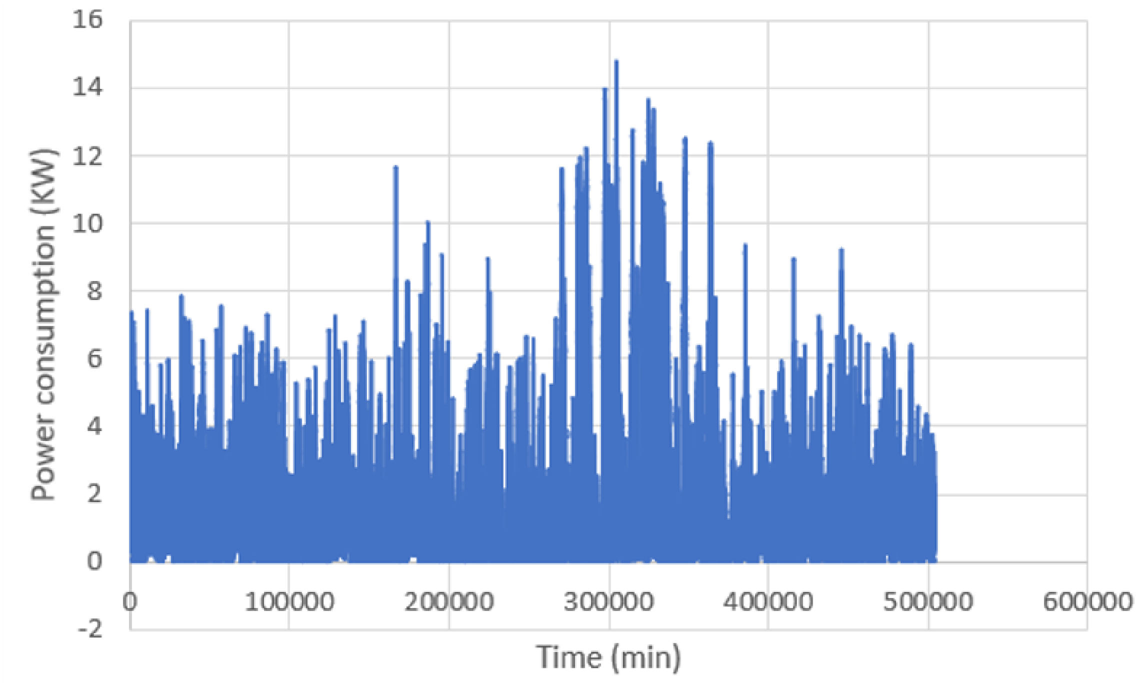

Figure 3.2: Data set representation for total use

The main reason for the generation of the maximum load curves as shown in Figure 3.3 was to have an improved idea about the load patterns, as the graph plotted with the whole dataset will not give a better idea. Data used here are time series data and it is of continuous values in minutes. From the continuous curve the different point like maximum value, minimum value etc are hard to find out.

The labelled dataset is graphically represented with maximum load curves for power consumption (KW) against time (min). The consumption of different parts of a residential building like living room as shown in Figure 3.4 , office room as shown in Figure 3.5.

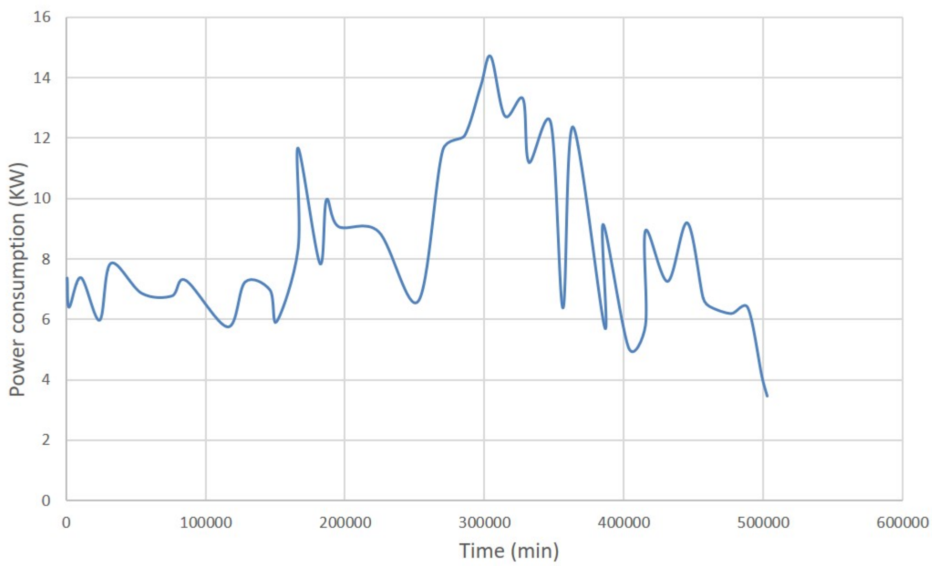

Figure 3.3: Max load curve for total use

The load curve for kitchen as shown in Figure 3.6 are graphically represented, so as to observe the load pattern. These factors are analysed to find out the correlation with load data and the most correlated features are selected. Each individual appliances had a max load curve plotted along with time.

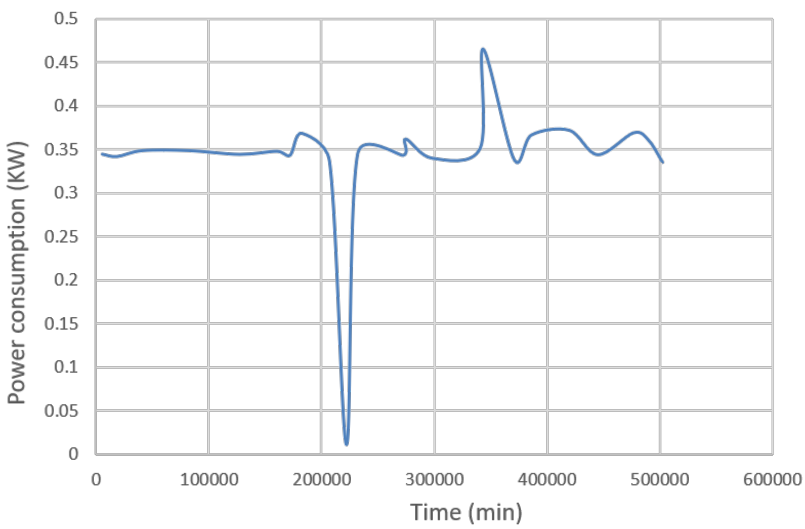

Figure 3.4: Maximum load curve for living room

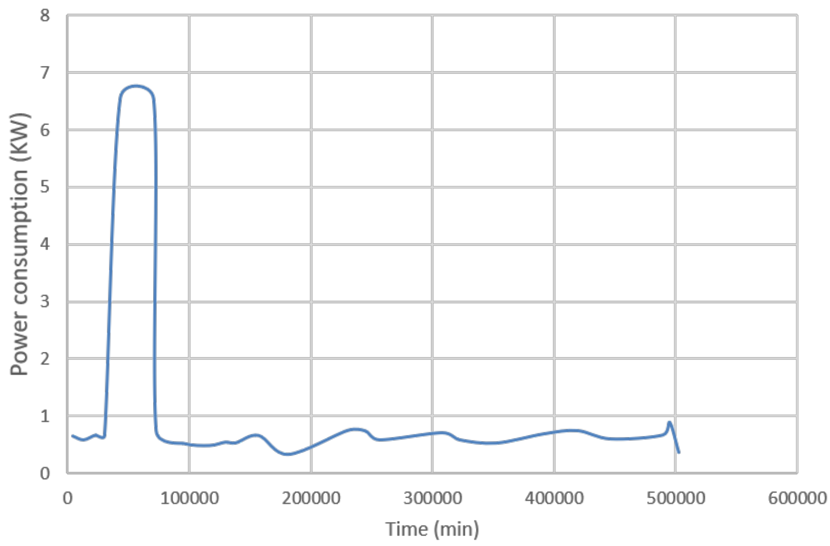

Figure 3.5: Maximum load curve for home office

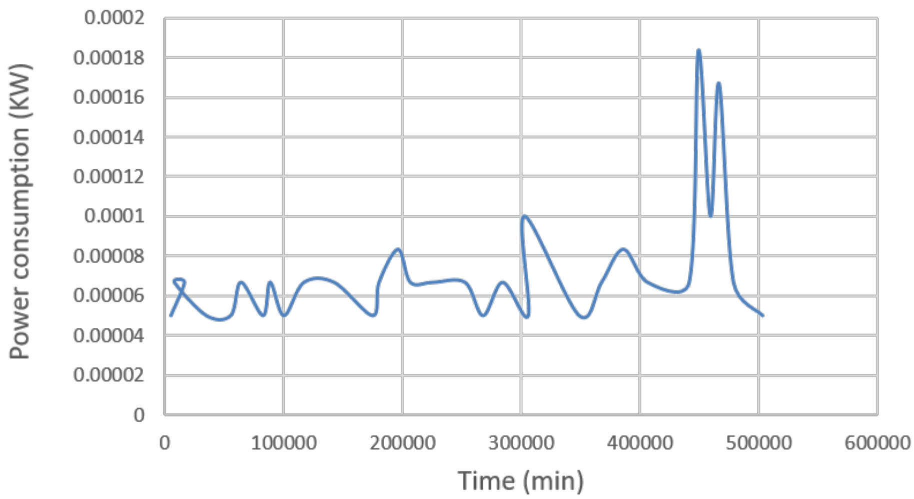

Figure 3.6: Maximum load curve for kitchen

3.4 Regression

It is the method by which an independent variable is used to find out the dependent variable. It mainly consists of two variables, a dependent and an independent variable which are used for the predictions. Linear regression model generates a linear equation having one or more independent variables and a dependent variable. Regression models can be used for finding the relation among different variable, to predict trends and for forecasting or impact of changes. If there is only one independent variable, then it is called as simple regression and if multiple independent variables are used for the predication of the dependent variable, then it is called as multiple regression.

3.5 Flow chart

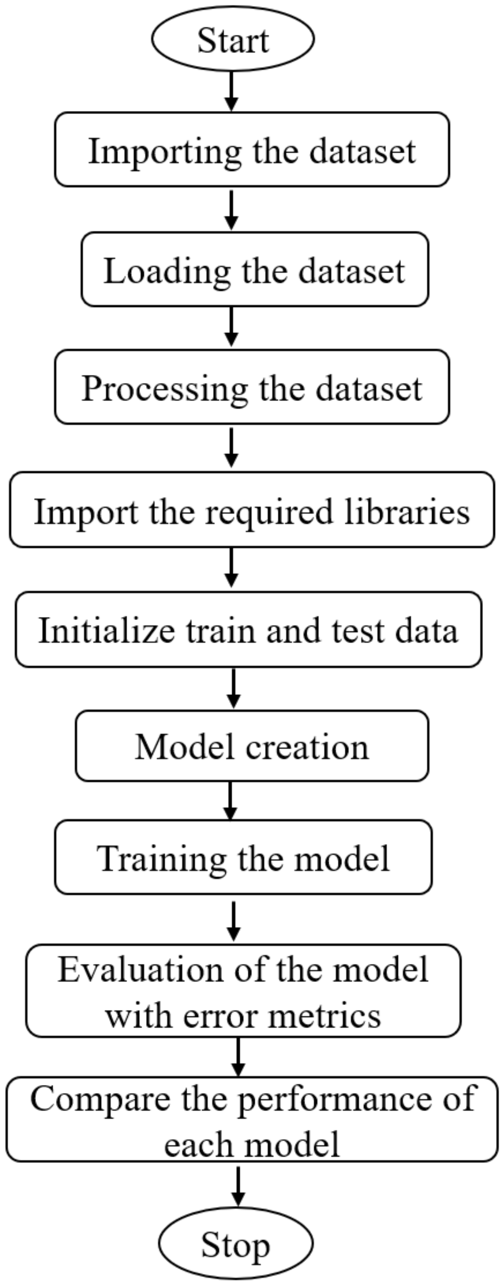

Figure 3.7: Flowchart of the proposed method

The Figure 3.7 gives an idea about the generalized flowchart for all the three methods used for the model creation. The only difference is in the layers used as it changes according to the respective model. The first step is to import the dataset with values of around 350 days over a time span of one minute. The dataset used contains time-series readings, where the variables have a sequence or it is continuous data obtained from daily observations of a smart meter. Then the CSV file is imported and read to find out the length of the data. The total range of the data is found out and a new dataset is created with hourly values.As the raw data is difficult to be processed it is converted into its corresponding numeric representation. For each of the three methods a new dataset was created from the existing dataset were the new dataset had data in hours instead of minutes and it was more useful in showing the variations in the forecast.

The created dataset was then saved as a new CSV file into the drive and was used for the model creation. MinMax Scaler was used to normalize the data into a range between 0 to 1. It transforms the data by scaling it to a given range without changing the shape of the original data. The required libraries like NumPy, Pandas, Matplotlib, Math etc are imported. NumPy was used for python programming and for working with arrays. Pandas was used for data manipulation and analysis. Math was used for performing mathematical operations, whereas Matplotlib was used for data visualization and graphical plotting.The train and test set created was used for the training and testing purpose. About 70 percentage of the data was used as train data and remaining 30 percentage was used as test data. The created train and test data are then reshaped for the model creation.

Hyper parameters such as batch size, optimizer used, loss function, training epochs, number of layers stacked, number of neurons were carefully optimized to avoid over fitting and under fitting conditions. Model training was done for 40 epochs with a batch size of 10. Drop out layers can be used to avoid regularization and over fitting. Early stopping is another way to avoid over fitting and generalization. During the iterations it was observed that after a particular number of epochs, there was no change in the loss values so the epochs was changed accordingly. So, after setting the parameters as shown in Table 3.1 and the model was created for load forecasting. The model validation was then done using the performance evaluation metrics and the performances were compared to find out the optimised model.

Table 3.1: Hyper parameters

Parameter settings	Value
Batch size	10
Epoch	40
Activation function	Sigmoid
Loss function	Mean squared error
Model optimizer	Adam

3.6 Deep learning models

3.6.1 Recurrent Neural Network

Due to the memory-based architecture of recurrent neural network (RNN), it's used for time series-based data. RNN represented in Figure 3.8 can be useful specifically to find the signals of each appliance as it analyses each item based on past data. Dynamic changes in energy consumption are specific to each appliance and is the signature of the appliances. RNN can be used for time series prediction and language modelling. The output of a state is given back to the input layer and retains all states previously encountered. So, every element considers the current state and previously occurred state, which helps to learn long-term dependence and make prediction by processing the sequence of events.

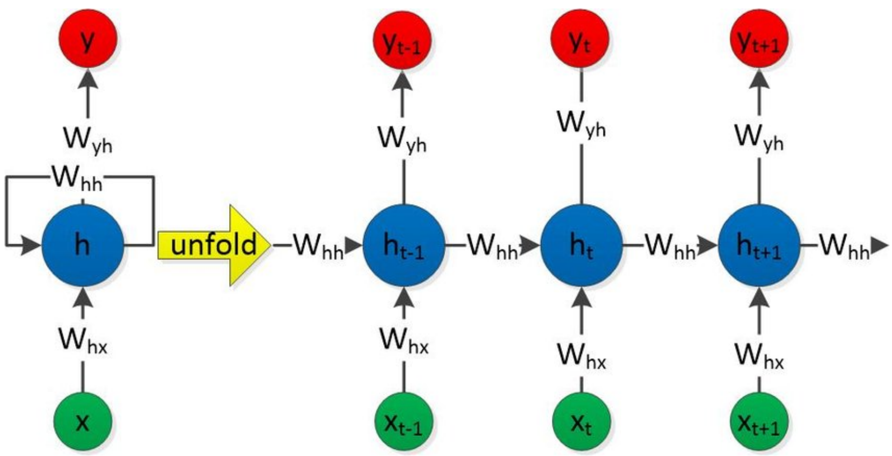

Figure 3.8: RNN architecture

The only problem of RNN is that as the dataset becomes large, the training becomes slow and learning capacity is weakened. RNN consist of a layer of memory cells that stores the data along the network. The disadvantages of RNN are vanishing gradients and exploding gradients. As these gradients carries the information, sometimes it carries excessive information which has errors or it will be very small. Both high and small parameters updates are not good for the neural network model.

3.6.2 Long Short-Term Memory

In order to overcome the challenges of RNN, memory cells that reserve a state and retains a carry in order to reduce the data damage throughout sequence processing are used and are called Long Short-Term Memory (LSTM). The chain-like shape consists of numerous interactive neural network layers that choose the quantity of information to be kept, the importance of the information to be recollected and portion of the memory cell which influences the output at the specified timestep. RNNs performs fine with sequential dataset, as they permit neuron networks in the identical layer to neural network. To active RNN the input given is the mains reading and a single valued output is produced, which is the load used by the appliance tested. To get the better of vanishing gradients issue the network uses units of LSTM and keep values on the built-in memory cells. LSTM as shown in Figure 3.9 are now being proposed as a robust solution for load forecasting. LSTM is a special kind of RNN capable of learning long-term dependencies. It is made up of three gates an input gate, an output gate and a forget gate. The gates control the amount of memory to be exposed.

3.6.3 Forget Gate

It is used to optimize the model by controlling the amount of information in the cell step. Unwanted informations form the previous state are removed if its of less importance to the current state.

3.6.4 Input Gate

New informations are added to the input gate consisting of sigmoid and tanh functions(activation functions).

3.6.5 Output Gate

It determines the value of the next hidden state.

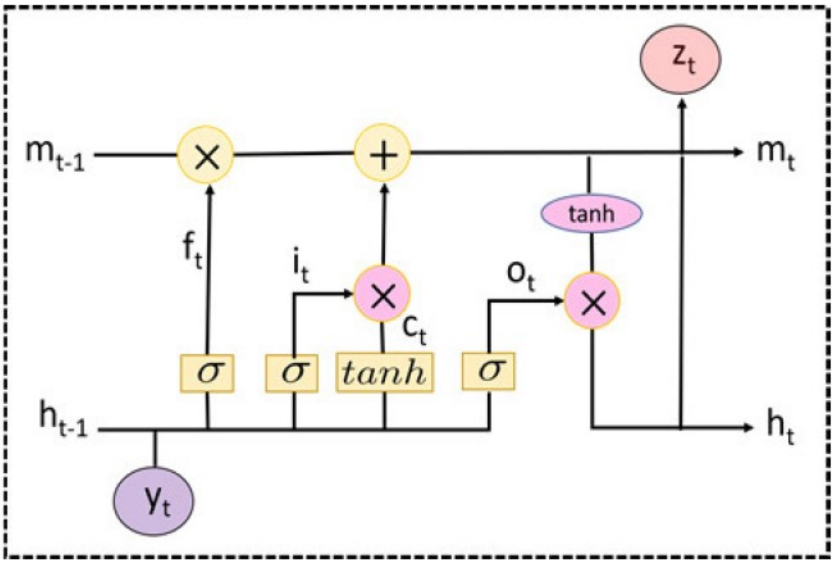

Figure 3.9: LSTM architecture

3.6.6 Bidirectional Long-Short Term Memory

Bi-LSTM is more advanced in comparison to uni-directional LSTM models. The Bi-LSTMs process the inputs in both the forward and backward directions as shown in Figure 3.10. In the forward direction it goes from past inputs to future inputs and in the backward direction, from future inputs to past inputs. The information from the past and future inputs are conserved through using different hidden layers. The output is then passed to the only similar output layer, which permits the Bi-LSTMs to preserve the context and data patterns well from both past and future inputs without delay. Since the inputs are processed in forward and backward directions, it performs better forecasts and classifications compared to uni-directional LSTMs in diverse areas. The main benefits of this model are:

1. To exploit the data features to extract the bidirectional temporal dependencies from available data.
2. To preserve the information from past and future inputs in the hidden states of LSTM cells.

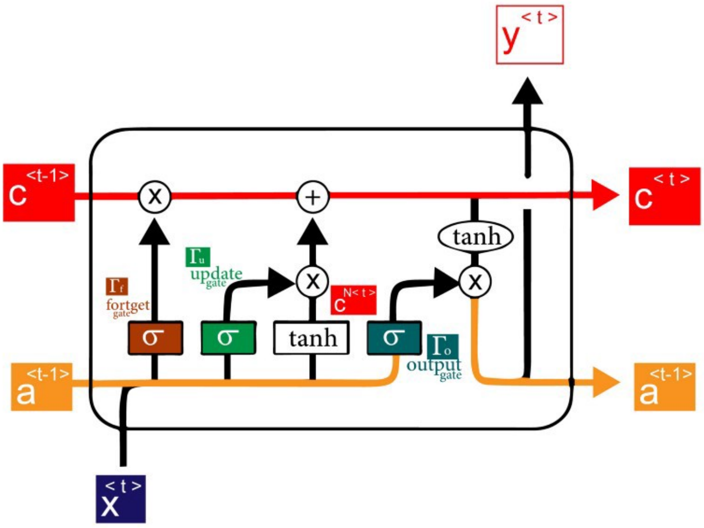

Figure 3.10: Bi-LSTM architecture

3.6.7 Gated Recurrent Unit

In GRU there is only a gate called the update gate, which is a combination of both the input and the forget gate. Also, the hidden state is used instead of cell state which reduces the number of system parameters as it reduces the gates and states. All these contributes to faster results from the model. GRU consist of 1D convolutional layer which extracts the information from the input raw data and helps in the improvement in model performance as shown in Figure 3.11. Bidirectional layers used to improve the performance and analyses the time series forward and backward. Multitask gated recurrent network (M-GRU) is used to execute energy disaggregation and detection of load status at the same time.

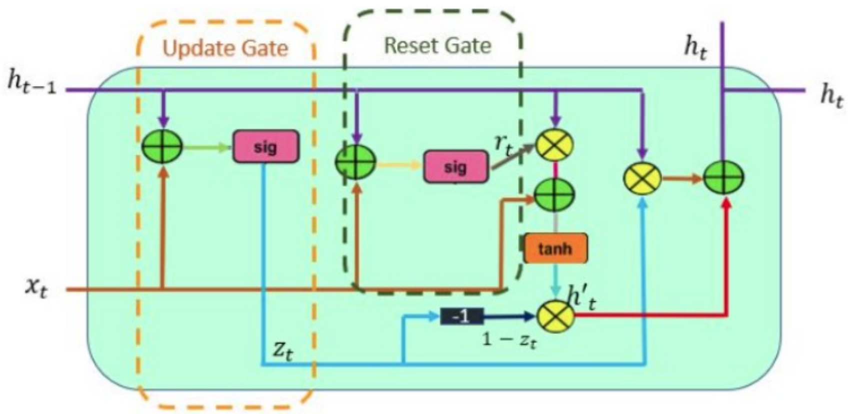

Figure 3.11: GRU architecture

3.7 Performance evaluation metrics

To build a generalized model we need to evaluate the model on different metrics so as to optimize the model performance and to obtain a better result. Performance evaluation metrics commonly used for load forecasting are Mean Absolute Error (MAE), Mean Square Error (MSE), Root Mean Square Error (RMSE), R-squared score. The different performance metrics are evaluated to find the model accuracy.

3.7.1 Mean Absolute Error

MAE is the difference among actual and predicted values and it should be minimum. This difference is called error. It is the most commonly used evaluation metric. Here $\widehat{y_i}$ is the predicted value, y_i is the actual value and N is the total number of values in the test set.

$$\mathbf{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \widehat{y_i}| \tag{3.1}$$

3.7.2 Mean Square Error

It is similar to MAE but the squared error between actual and predicted values are found out. Squaring is done to avoid cancellation of negative terms and it is the main advantage over MAE.

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y_i})^2 \tag{3.2}$$

3.7.3 Root Mean Square Error

It is the square- rooted value of MSE and is the most famous evaluation metric for regression models. It is used more than MSE, as sometimes its values are too high to compare and for easier interpretation, we are taking the square root for bringing the error values to the same level of prediction errors.

$$\mathbf{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y_i})^2} \tag{3.3}$$

3.7.4 R-squared

R-squared gives an idea about how well our model performance and how well the fits the dependent variables. It is also known as coefficient of determination.

$$\mathbf{R}^2 = \mathbf{1} - \frac{\sum_i^N (y_i - \hat{y}_i)^2}{\sum_i^N (y_i - \overline{\mathbf{y}}_i)^2} \tag{3.4}$$

3.8 Conclusion

This chapter addressed the different stages involved in the model creation. Block diagram representation of the model and the flow chart was also explained in detail. The different learning models are mentioned along with the performance evaluation metrics used for model validation.

Chapter 4

RESULTS AND ANALYSIS

4.1 Introduction

The platform used for the model generation was opensource platform TensorFlow employed in background and keras, as it is used for deep learning tasks. The model training and evaluation was done using Google Colab. It works on cloud and provides free GPUs. The model was bulit using Python programming.

4.2 LSTM model

The dataset used contains smart meter readings of a household with the total consumption and individual appliances consumption readings. The input data had around 503911 readings of 350 days over a time span of 1 minute. The dataset was then preprocessed into a new data set with readings in hours. So the total readings in hours was found from the original dataset to specify the range of values. The model creation was done using this new dataset and splitting it into test and train data after normalisation. 70 percentage of the data goes for training and the rest 30 percentage goes for testing. Different Python libraries were imported and the reshaped train data was used for model creation and test data for validation. Sequential model with LSTM and dense layers are used with the input shape specified, so that the data is passed into the next layer. Sigmoid activation function was used with mean squared error as loss function and Adam optimizer. The model training was done for 40 epochs with a batch size of 10. The prediction model was plotted graphically with power consumption (KW) vs time (H). Two different graphs were plotted in orange and blue colors as shown in Figure 4.1. The blue color represents the actual values (original dataset) and the orange color represents the predicted values.

The output plot gives a comparison of true values against the predicted values. The independent axis in the figure represents the time of the day and the dependent axis indicates the energy usage. From the graph it's observed that the predicted values closely follow the actual values, which shows good accuracy and low error of the proposed deep forecasting models.In the output graph the time ranges for about 150 hours and the power consumption for about 200 KW. The R-squared values are found out evaluate the model performance. Is also called as coefficient of determination and the values lie in the range 0 to 1. Higher the R-squared value better will be the model performance. In this work the R-squared value is 0.839, which is good as it is near to 1. In regression problems we use evaluation metrics to find out how close are the predictions to the actual value.So the model performance are determined based on these metrics.

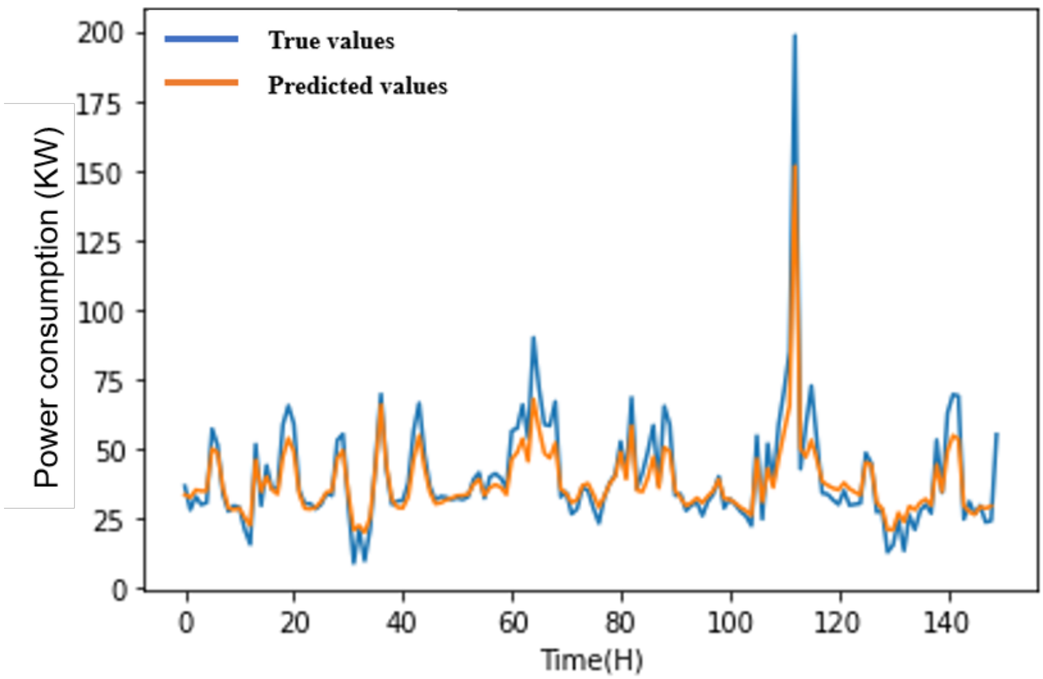

Figure 4.1: Forecast results for LSTM model

Actual and forecasted load values for LSTM model are tabulated in Table 4.1. The table gives an idea about how close are the predicted values to the actual values.In case of regression the relationship between independent and dependent variables are found out and the values are never exactly the same, but close to the actual values. So from the table it is observed that the predicted values are close to the actual values.

Table 4.1: Comparison table for LSTM model

LSTM	
Actual load values (KW)	**Forecasted load values (KW)**
36.33	36.131
27.832	35.51
32.615	40.104
29.688	38.42
30.351	38.096
56.901	53.257
51.081	48.326
33.394	34.384
27.243	25.866
29.283	24.917
28.687	25.837
20.665	22.843
15.42	21.211
51.384	52.121
29.321	32.833
43.810	42.308
35.323	33.835
35.543	32.507
58.427	48.346
65.254	54.086

4.3 Bi-LSTM model

It is a combination of two RNN netwoks.In Bi- LSTM input is processed in both directions, both in forward and backward directions. So Bi- LSTM model gives a better prediction of data compared to LSTM, as more information is available to the network. Similar to LSTM the BiLSTM model the the dataset is imported from Google Drive and was loaded. The dataset was preprocessed using MinMax Scaler and the values were converted into a range of 0 to 1. The features of the data are scaled to a given range without changing the original shape. Different libraries were imported for model creation. The length and test data and train data are found out so as to split the original dataset into two. The input data was then reshaped and the sequential model was created. The different layers used are bidirectional layers and dense layers. The bidirectional layers connects two opposite hidden layers to the output layer, such that output layer gets information from the past and future states at the same time. Sigmoid activation function was used with mean squared error as loss function and Adam optimizer. The model training was done for 40 epochs with a batch size of 10.

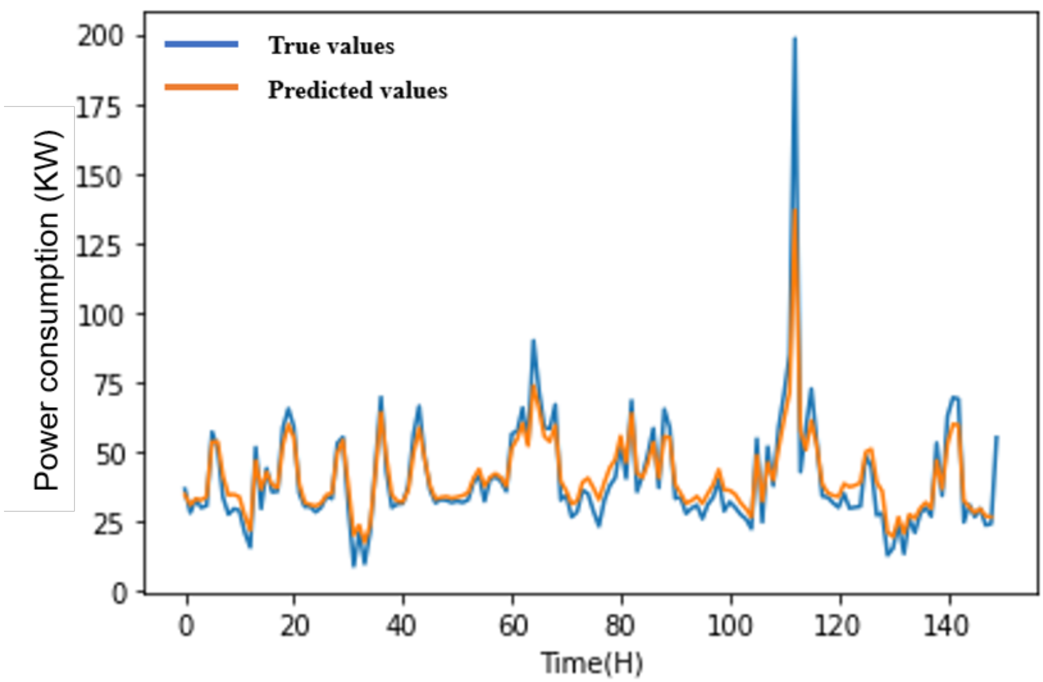

Figure 4.2: Forecast results for Bi-LSTM model

The prediction model was plotted graphically with power consumption (KW) vs time (H). Two different graphs were plotted in orange and blue colors as shown in Figure 4.2. The blue color represents the actual values (original dataset) and the orange color represents the predicted

Table 4.2: Comparison table for Bi-LSTM model

Bi-LSTM	
Actual load values (KW)	**Forecasted load values (KW)**
36.33	33.326
27.832	25.959
32.615	29.795
29.688	28.639
30.351	29.682
56.901	51.227
51.081	51.532
33.394	39.919
27.243	31.298
29.283	29.502
28.687	27.831
20.665	21.264
15.42	17.773
51.384	41.899
29.321	31.527
43.810	40.210
35.323	37.097
35.543	35.902
58.427	50.444
65.254	56.096

values. From the graph it is observed that the prediction was better compared to the LSTM model.The R-squared value was found to be 0.86137, which is good as it is close to 1. Higher the R-squared value better are the predictions.

Actual and forecasted load values for LSTM model are tabulated in Table 4.2. From the table it is observed that the predicted values are closer to the actual values and gives a better result compared to the previous model.

4.4 GRU model

GRU is similar to RNN, with better computation efficiency. All the process mentioned in the previous two model are repeated. The main difference is that the sequential model was built using GRU layers and dense layers. Mean squared error was used as loss function and the optimizer used was Adam . The model training was done for 40 epochs with a batch size of 10.

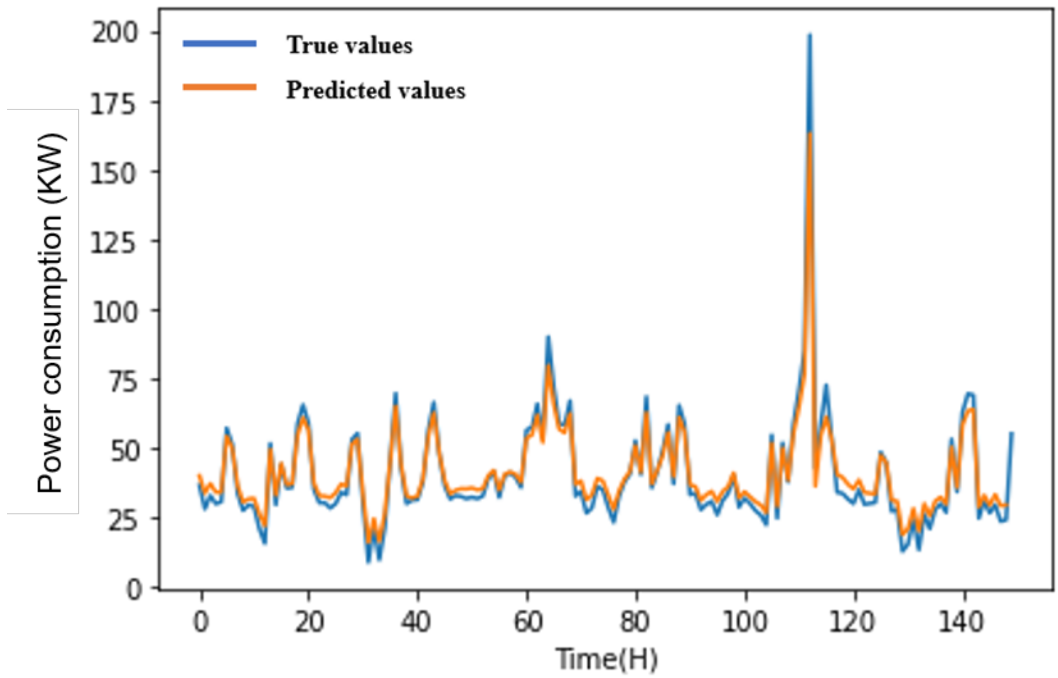

Figure 4.3: Forecast results for GRU model

GRU only has two gates, the update gate and reset gate.The update gate determines the amount of information passed to the next gate and the reset gate determines the amount of unwanted data to be removed. From the graph it is observed that the model gives a better prediction compared to the other two models. The R-squared value is 0.8837, which is much higher and GRU model gives the best prediction as shown in Figure 4.3. It has a simple architecture and

Table 4.3: Comparison table for GRU model

GRU	
Actual load values (KW)	**Forecasted load values (KW)**
36.33	39.769
27.832	33.516
32.615	36.96
29.688	33.842
30.351	33.836
56.901	53.948
51.081	49.834
33.394	36.177
27.243	30.295
29.283	31.4
28.687	31.599
20.665	26.009
15.42	21.832
51.384	49.219
29.321	32.736
43.810	44.191
35.323	36.582
35.543	36.871
58.427	54.399
65.254	60.599

gives faster computations as it uses less memory. Also it uses less training parameters compared to LSTM and Bi-LSTM model.

Actual and forecasted load values for LSTM model are tabulated in Table 4.3. From the table it is observed that the predicted values are closer to the actual values and gives a better result compared to the previous model. The GRU model had the highest value for R-sqaure, which gives an idea about model accuracy. The R-sqared value of GRU is 0.88 whereas its 0.86 for Bi-LSTM and 0.83 for LSTM.

4.5 Performance evaluation

Accuracy of the forecasting models are of great importance, as decisions are made based on the forecast. So a comparison study between the different deep learning models was done to find out the best model for household load forecasting. The values of evaluation metrics are compared to evaluate the performance of forecasting models. Forecasting are done based on known time-series data data and the ability of a model to predict the unknown future depends on its performance.The commonly used evaluation metrics are R-squared, MAE, MSE and RMSE.

4.5.1 R-squared

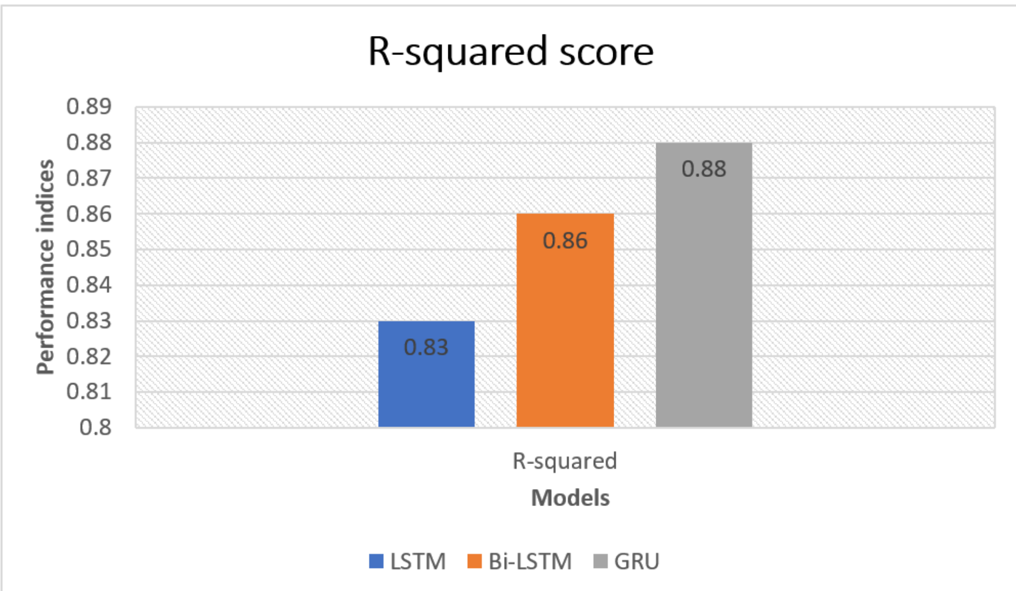

Figure 4.4: R squared comparison of models

R-squared gives an idea about how much variance of the dependent variable can be explained by the variance of independent variable. Figure 4.4 shows the graphical representation of R-squared values for different models. Higher the value better the forecast and it ranges from 0 to 1. Mostly values close to 1 are preferred for a good model. R-squared are not ideal in case of too many independent variables as it does not consider over fitting problems. Here the R-squared values are 0.83 for LSTM, 0.86 for Bi-LSTM and 0.88 for GRU. So by comparing the values, the GRU model has the highest score.

4.5.2 MAE

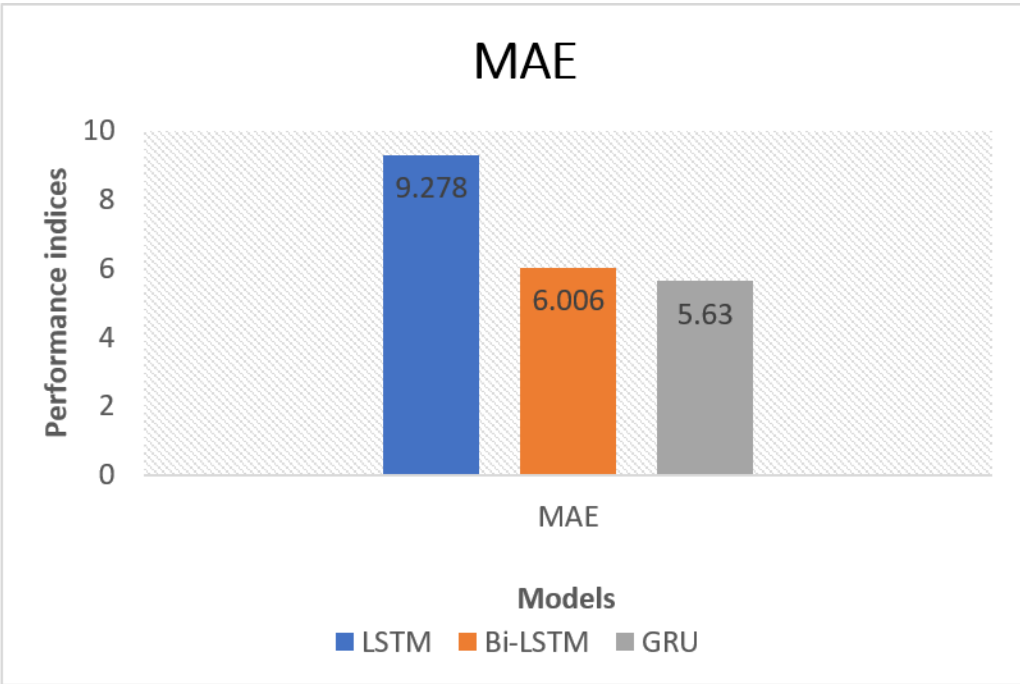

Figure 4.5: MAE comparison of models

MAE gives the average of absolute error, were the error is the difference between actual value and forecasted value. Lower the MAE value, better the model. Also it does not an idea about the size error but are compared with other metrics to find if the errors are higher. Here MAE values for LSTM model is 9.27 , 6.00 for Bi-LSTM and 5.63 for GRU. From the comparison it is observed that the GRU model has the lowest value. Figure 4.5 shows the graphical representation of MAE values for different models.

4.5.3 MSE

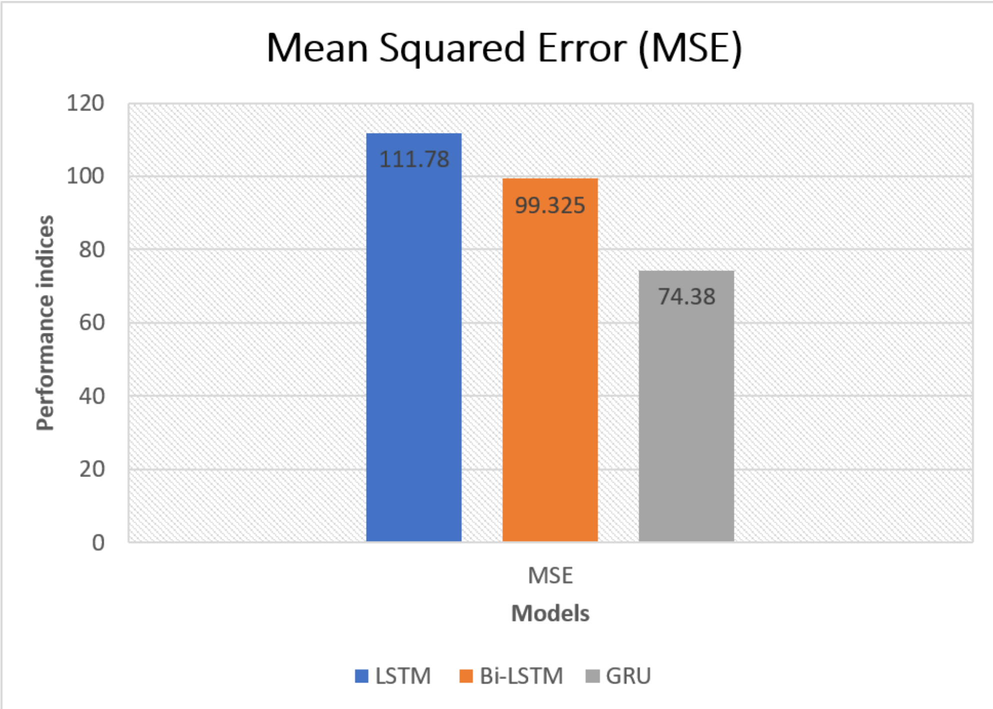

Figure 4.6: MSE comparison of models

It is the average of error squares. The quality of the forecasting models are evaluated by using MSE. The MSE value should be minimum and it gives an idea about the accuracy of the model. Here Figure 4.6 shows the graphical representation of MSE values for different models, were MSE value of LSTM is 111.78, 99.32 for Bi-LSTM and 74.38 for GRU. So the GRU model has the lowest value.

4.5.4 RMSE

It is the square root of MSE. Lower values are desired for better performance. The RMSE values as shown in Figure 4.7 has the same unit of forecasted value, which makes it easier to understand than MSE. It is used to make the squared values back to the initial state. Also it can be compared with MAE value to determine whether the forecast contains errors. LSTM model has the highest value for RMSE with 10.57 and the lowest for GRU with 8.62 and 9.96 for Bi-LSTM.

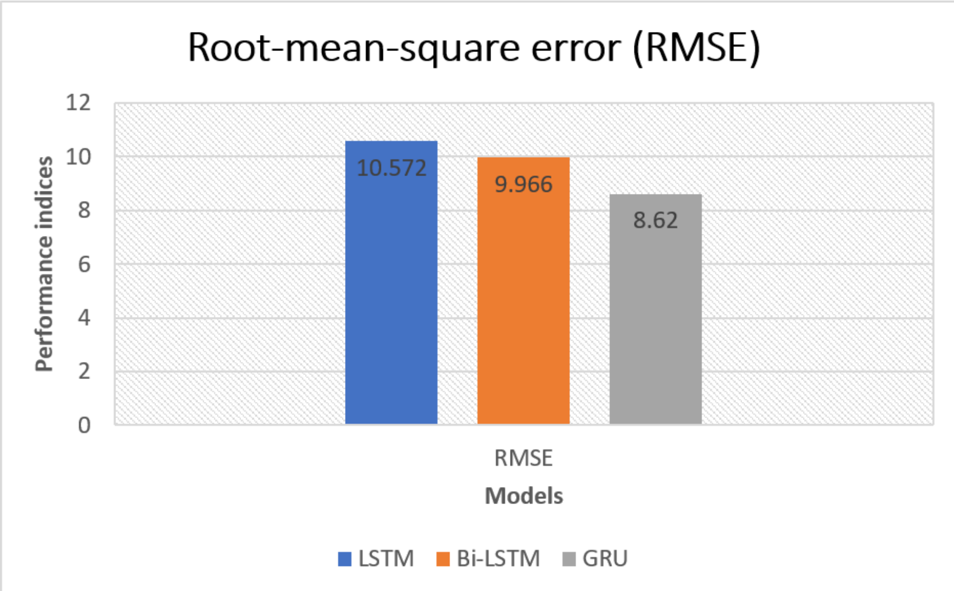

Figure 4.7: RMSE comparison of models

Table 4.4: Model performances

Training Model	Household energy dataset			
	R-squared	MAE	RMSE	MSE
LSTM	0.839229	9.278283	10.572885	111.78591
B-LSTM	0.861372	6.006122	9.966195	99.32505
GRU	0.883737	5.638088	8.624935	74.38952

In the Table 4.4 the model performances are tabulated and the most suitable method was found out to be the GRU model. The model accuracy was based on the highest R-squared value and with least error values for evaluation metrics like MAE, MSE and RMSE.

4.6 Conclusion

The results obtained for different load forecasting models such as LSTM, Bi-LSTM and GRU are evaluated based on performance metrics. The forecasted load values were then tabulated against the actual load values, so as to give a better idea about the prediction models. The performance comparison of various methods was done and the one with better accuracy and least error was finalized for load forecasting in households using smart meter data.

Chapter 5

CONCLUSION AND FUTURE SCOPE

Load forecasting in households is very important as it helps to reduce the energy usage and improve the utilization efficiency. More accurate models are created so as to improve the smart grid applications and to meet demand side management. It can be used to improve the economic and environmental factors and helps the power utility companies to reduce the cost of production and raw materials. This work deals with a novel approach for load forecasting using a household dataset containing reading around 350 days over a span of one minute. Load forecasting results depend upon the accuracy of the data inputted, so the best features that are most correlated are chosen to avoid uncertainty. Load forecasting can be done using various methods like statistical approaches and new methods like deep learning techniques. Here models like LSTM, Bi-LSTM and GRU are the main deep learning methods used and their performance evaluation was done based on error metrics like R-squared, MAE, MSE and RMSE to find out the best model. After a comparison study using different error metrics, the GRU model was found to have the highest value for R-squared and lower values for MAE, MSE and RMSE. So it was found that GRU was the most suitable model for load forecasting. Future works can be done by using hybrid models for finding out medium and long term load forecasting. Furthermore the forecasting accuracy and efficiency can be improved by combining various forecasting methods to develop a robust and stable model.

References

[1] D. Syed, H. Abu-Rub, A. Ghrayeb, and S. S. Refaat, "Household-level energy forecasting in smart buildings using a novel hybrid deep learning model," *IEEE Access*, vol. 9, pp. 33 498–33 511, 2021.

[2] I. IHSANE, L. MIEGEVILLE, N. Aït-AHMED, and P. GUERIN, "New evaluation metrics for electrical demand forecasting: Application to the residential sector," in *2018 AEIT International Annual Conference*. IEEE, 2018, pp. 1–6.

[3] C. Cui, M. He, F. Di, Y. Lu, Y. Dai, and F. Lv, "Research on power load forecasting method based on lstm model," in *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*. IEEE, 2020, pp. 1657–1660.

[4] M. J. A. Shohan, M. O. Faruque, and S. Y. Foo, "Forecasting of electric load using a hybrid lstm-neural prophet model," *Energies*, vol. 15, no. 6, p. 2158, 2022.

[5] M. A. Devlin and B. P. Hayes, "Non-intrusive load monitoring and classification of activities of daily living using residential smart meter data," *IEEE transactions on consumer electronics*, vol. 65, no. 3, pp. 339–348, 2019.

[6] T. Hossen, A. S. Nair, R. A. Chinnathambi, and P. Ranganathan, "Residential load forecasting using deep neural networks (dnn)," in *2018 North American power symposium (NAPS)*. IEEE, 2018, pp. 1–5.

[7] J. Jeyaranjani and D. Devaraj, "Deep learning based smart meter data analytics for electricity load prediction," in *2019 IEEE International Conference on Clean Energy and Energy Efficient Electronics Circuit for Sustainable Development (INCCES)*. IEEE, 2019, pp. 1–5.

[8] X. Tang, Y. Dai, T. Wang, and Y. Chen, “Short-term power load forecasting based on multi-layer bidirectional recurrent neural network,” *IET Generation, Transmission & Distribution*, vol. 13, no. 17, pp. 3847–3854, 2019.

[9] X. Guo, Y. Gao, Y. Li, D. Zheng, and D. Shan, “Short-term household load forecasting based on long-and short-term time-series network,” *Energy Reports*, vol. 7, pp. 58–64, 2021.

[10] E. Aguilar Madrid and N. Antonio, “Short-term electricity load forecasting with machine learning,” *Information*, vol. 12, no. 2, p. 50, 2021.

[11] F. L. Quilumba, W.-J. Lee, H. Huang, D. Y. Wang, and R. L. Szabados, “Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities,” *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 911–918, 2014.

[12] L. Pereira, D. Costa, and M. Ribeiro, “A residential labeled dataset for smart meter data analytics,” *Scientific Data*, vol. 9, no. 1, pp. 1–11, 2022.

[13] R. Jiao, S. Wang, T. Zhang, H. Lu, H. He, and B. B. Gupta, “Adaptive feature selection and construction for day-ahead load forecasting use deep learning method,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4019–4029, 2021.

[14] J. Zhao and J. Huang, “Application and research of short-term power load forecasting based on lstm,” in *2020 International Conference on Information Science, Parallel and Distributed Systems (ISPDS)*. IEEE, 2020, pp. 212–215.

[15] M. D’Incecco, S. Squartini, and M. Zhong, “Transfer learning for non-intrusive load monitoring,” *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1419–1429, 2019.

[16] H. Jahangir, H. Tayarani, S. S. Gougheri, M. A. Golkar, A. Ahmadian, and A. Elkamel, “Deep learning-based forecasting approach in smart grids with microclustering and bidirectional lstm network,” *IEEE Transactions on Industrial Electronics*, vol. 68, no. 9, pp. 8298–8309, 2020.

[17] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, and Y. Xu, “Short-term residential load forecasting based on resident behaviour learning,” *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 1087–1088, 2017.

[18] Z. Yu, Z. Niu, W. Tang, and Q. Wu, “Deep learning for daily peak load forecasting–a novel gated recurrent neural network combining dynamic time warping,” *IEEE access*, vol. 7, pp. 17 184–17 194, 2019.

[19] P. Jia and Y. Miao, “Research on short-term forecast of power load based on adam-gru,” in *Journal of Physics: Conference Series*, vol. 1693, no. 1. IOP Publishing, 2020, p. 012116.

[20] M. Razghandi, H. Zhou, M. Erol-Kantarci, and D. Turgut, “Short-term load forecasting for smart home appliances with sequence to sequence learning,” in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[21] L. Sehovac and K. Grolinger, “Deep learning for load forecasting: Sequence to sequence recurrent neural networks with attention,” *IEEE Access*, vol. 8, pp. 36 411–36 426, 2020.

[22] G. Gong, X. An, N. K. Mahato, S. Sun, S. Chen, and Y. Wen, “Research on short-term load prediction based on seq2seq model,” *Energies*, vol. 12, no. 16, p. 3199, 2019.

[23] J. Shi, T. Tan, J. Guo, Y. Liu, and J. Zhang, “Multi-task learning based on deep architecture for various types of load forecasting in regional energy system integration,” *Power system technology*, vol. 5, no. 3, pp. 698–707, 2018.

List of Publication

[1] Sanjay Steephen, R Sheeba and N Naufal," A Review on Non-Intrusive Load Monitoring Using Deep Learning", *International Conference on Communications and Cyber-Physical Engineering (ICCCE)*,2022. (Accepted)