FORECASTING OF SOLAR IRRADIATION USING DEEP LEARNING ALGORITHMS.

A DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF
MASTER OF TECHNOLOGY
IN
POWER SYSTEM

SUBMITTED BY
ROMAN KUMAR JHA
2K20/PSY/16

UNDER THE SUPERVISION OF
DR. MADAN MOHAN TRIPATHI

DEPARTMENT OF ELECTRICAL ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(FORMERLY DELHI COLLEGE OF ENGINEERING)
BAWANA ROAD, DELHI-110042
MAY, 2022

Department of Electrical Engineering
Delhi Technological University
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Roman Kumar Jha,2k20/PSY/16 student of MTech (Power System), hereby declare that the project dissertation "Forecasting of Solar Irradiation using Deep Learning Algorithms" which is submitted by me to the department of Electrical Engineering, Delhi Technological University, Delhi, in partial fulfilment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation.

Roman Kumar Jha.

Place: Delhi
Date:31/05/2022

Roman Kumar Jha

Department of Electrical Engineering
Delhi Technological University
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

<u>CERTIFICATE</u>

I hereby certify that the project report titled “Forecasting of Solar Irradiation using Deep Learning Algorithms” which is submitted by Roman Kumar Jha, 2K20/PSY/16 student of MTech, Department of Electrical Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision.

Place: Delhi

Date:31/05/2022

Dr. Madan Mohan Tripathi
(Professor)
(SUPERVISOR)

ABSTRACT

This dissertation presents one of many application of Machine Learning (ML) and Deep Learning in the field of forecasting. ML algorithms used Multivariate Linear Regression(MLR), Support Vector Regression (SVR), Feed Forward Neural Network(FFNN) and Layered Recurrent Neural Network(RNN) to make solar irradiation forecasting. The forecasting has been done for the period of ten months in 2021 based on the historical data available for the year 2019 and 2020.MATLAB has been used to develop the ML model. The model developed using the above mentioned algorithms have been compared on the basis of key performance indicators(KPI). The indicators used are mean square error(MSE), Root Mean Square Error(RMSE), Mean Absolute Error(MAE), Mean Absolute Percentage Error(MAPE) and R Square Value (coefficient of determination). This dissertation proposes forecasting of solar irradiation using deep learning algorithm. The algorithm used in this dissertation is sequence to sequence (S2S) algorithm which uses LSTM cell in its encoder and decoder sections. Hence the forecasting has initially been done with LSTM (Long Short Term Memory) in order to make a comparative analysis. Python has been used for developing Deep Learning models. The algorithms are developed on Jupyter notebook using Keras and Tensor flow libraries. The loss function used for optimisation is Mean Square error (MSE) and the key performance Indicator is Root Mean Square Error (RMSE). The results obtained are well within the desirable limits for both ML and Deep Learning.

ACLKNOWLEDGEMENT

I sincerely thank Dr. Madan Mohan Tripathi, Professor Department of Electrical Engineering for sparing his valuable time and guiding me throughout the course. He has always encouraged me to work on trending technologies with respect to electrical engineering. His valuable guide has helped me to evolve over a period of time and add the knowledge of ML and Deep Learning in understanding it's application in Electrical Engineering.

I would also like to thank Dr. J. N. Rai, Professor, Department of Electrical Engineering for taking the course 'Forecasting Techniques in Power System' in 1st semester of the course. It has immensely helped in completing the thesis work. I would also like acknowledge the Major Project I evaluation committee for their thoughtful insights.

Roman Kumar Jha .

ROMAN KUMAR JHA

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS, ABBREVIATIONS

SIR Solar Irradiation

HDD Heating Degree Day

CDD Cooling Degree Day

THI Temperature Heating Index

WCI Wind Chilling Index

AI Artificial Intelligence

ML Machine Learning

AR Auto Regressive

MA Moving Average

ARMA Auto Regressive Moving Average

ARIMA Auto Regressive Integrated Moving Average

ARIMAX Auto Regressive Integrated Moving Average with Explanatory Variables

SVR Support Vector regression

FFNN Feed Forward Neural Network

RBFNN Radial Belief Feed Forward Neural Network

MLP Multilevel Perceptron

DRNN Deep Recurrent Neural Network

LSTM Long Short Term Memory

GRU Gated Recurrent Unit

DCNN Deep Convolutional Network

DBN Deep Belief Network

RBN Radial Basis Network

SAE Stacked Auto Encoder

GAN Generative Adversarial Network

S2S Sequence to Sequence

MAE Mean Absolute Error

RMSE Root Mean Square Error

MAPE Mean Absolute Percentage Error.

DBSCAN Density-Based Spatial Clustering of Applications with Noise

PCA Principle Component Analysis

SNE Stochastic Neighbour Embedding

Norm Normalisation

Max Maximum

Min Minimum

Re LU Rectified Linear Activation

LM Levenberg Marquardt Algorithm

BR Bayesian regularisation algorithm

SCG Scaled Conjugate Gradient Algorithm

Adam Adaptive Moment Optimizer

RMS Prop Root Mean Square Propagation

MLR Multivariate Linear Regression

DISCOM Distribution Company

KPI Key Performance Indicator

AT&C Aggregate Technical and Commercial

CHAPTER 1

INTRODUCTION

1.1 Forecasting in Power System

Forecasting in power system has become an important tool for planning, operation and control of system. Accurate model of forecasting has become essential for the operation and planning for the utility. It is being used in voltage control, power system protection & security assessment, condition monitoring, scheduling maintenance of power system network and demand forecasting. Forecasting are essential for the all the stakeholders in the power industry like energy suppliers, financial institutions, Generation, Transmission and Distribution companies. Forecasting can be done to predict Load/Demand, Price, Solar Irradiation. Solar Irradiation forecasting is usually a weather phenomenon as it's input parameters are weather variables. Solar Irradiation is the measure of sun's intensity which helps in the future prediction of energy generation from solar power installations from minutes to days ahead of time. Since there is unpredictability of solar Irradiation on different parts of the earth, predicting solar Irradiation shall prove to be an advantage for solar power generators. Exact prediction of solar irradiation can provide a meaningful guidance for solar operators and power planning.

There are important factors which influence forecasting. Among all the factors weather variables play an important role in forecasting of solar irradiation. Various weather variables could be considered for SIR. Weather variables can be used in the primitive form or as a modified form in the forecasting. Primitive variables are those which are directly obtained from the weather stations and are used for the forecasting purpose. Temperature in general is the most commonly used exogenous variable. Wind speed and humidity are other weather variables used in its primitive form. Some of the modified weather variables which are specially temperature derivative are HDD and CDD. These are the deviation of mean temperature from a comfortable temperature. Heat index which depends on the relative humidity also acts as modified variable. THI and WCI are also used in forecasting. THI is the measure of heat stress and similarly WCI is the cold stress in winter. There are time factors which also effects SIR. Hour of the day, day of the week and time of the year (season) are important variables to be considered while forecasting solar irradiation. Forecasted SIR along with orientation of solar panels can help in prediction of solar power.

The market for photovoltaics are ever-growing and taking climate change and international commitments in consideration, Government across the world are focussing on increasing the share of energy generation form renewable sources. India being a tropical country has a huge potential to generate solar power. According to Ministry of New and Renewable Energy, India has increased its share of renewable energy in its total installed generation capacity to 26.53 %. Solar capacity has increased from 2.6 GW to more than 46 GW. India is now at 4th global position for overall installed renewable energy capacity and aims to achieve 175 GW of solar power till 2022.

With increasing dependency on solar energy resources it becomes evident to integrate it to the electricity grid. One of the main challenge of solar energy is its intermittent and stochastic nature which poses numerous problems to the electricity grid operator. We can overcome the problem by using energy storage system and by having suitable forecasting techniques. The intermittent and stochastic production management bears a significant cost to utilities and employing forecasting methods can make a considerable reduction of cost. The uncertainty of supply from solar energy may cause a situation of imbalance between supply and demand of power. The balance between the production and the consumption is prerequisite in the dynamic power system to maintain the continuity. In case of loss of supply of solar power due to shadow of cloud and overproduction of solar power, this balance may be broken. Deficit in power supply may lead to reduction in frequency from the reference value. Overproduction on the other hand may increase the frequency level and present a danger for the electrical machines. Hence the intermittent supply of solar power poses a problem for the grid operators. This leads to continuously provide supply by other means of sources, which is however difficult to react immediately.

With respect to the above discussions, there arises a need for solar irradiation forecasting. Forecasting output power with the help of SIR is required for good operation of the power grid and for an optimal management of the energy flows occurring into the power system. It is necessary to estimate the reserves, scheduling the power system and congestion management for optimally managing the storage and for trading in the electricity market. Power reserves are classified into contingency reserves which are used in case of specific event such as power plant switch on and no event reserves which are used continuously for instance, because of unreliable load prediction. These reserves are started at various time scale, within 1 minute, from 1 minute to 1 hour and more than 1 hour. The forecasting of power helps in better anticipation and management. Forecasting can be done at various temporal horizons. Forecasting at time horizon allows storage management by decreasing the amount of flexible reserves and also optimises the management of energy storage by anticipating the charge and discharge cycle. The classification of forecasting on time scale [8],

1. Long Term load forecasting for 5 to 20 years

• used for understanding the capacity of power producing

• used for building new sub stations and transmission lines

• shall help in decision making whether to add new features in prevailing systems

• employing human resource

2. Medium Term load forecasting for few weeks to few months

• Used to meet necessities in extreme weather.

• Used for acquiring fuels and reviewing tariffs.

3. Short Term load forecasting from few hours to few weeks

• Used in Prediction calculation on hourly basis.

• helps in finding out upper limit of the demand of electricity.

• allocation of fuel to generation units, short term Maintenance, generator unit commitment.

4. Very Short Term load forecasting from minute to an hour

• Used in Energy Management Systems (EMS)

1.2 Forecasting Techniques in Power System

Over a last few decades' number of forecasting methods have been developed and many more will come into existence. Forecasting methods that have been used earlier was based on traditional mathematical models. However, over a period of time numerous modern techniques have been used and advent of AI and ML have only made the forecasting more accurate and time saving. Forecasting models can be of two types,

- Multivariate Forecasting: This type of forecasting takes into account the relationship between various influencing factors and values.
- Time Series Forecasting: This type of forecasting mostly considers historical series. This is easier and quicker as it eliminates complicated and non-objective factors. Some of the most widely used time series forecasting techniques are statistical, machine learning and hybrid methods.

This section presents a brief classification of forecasting models that are in use for short, medium and long term forecasting.

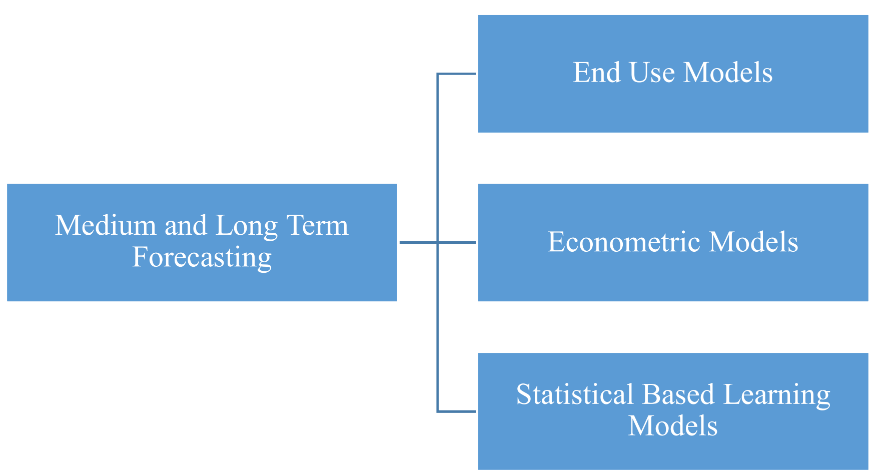


Figure 1: Forecasting models for Medium and Long term forecasts.

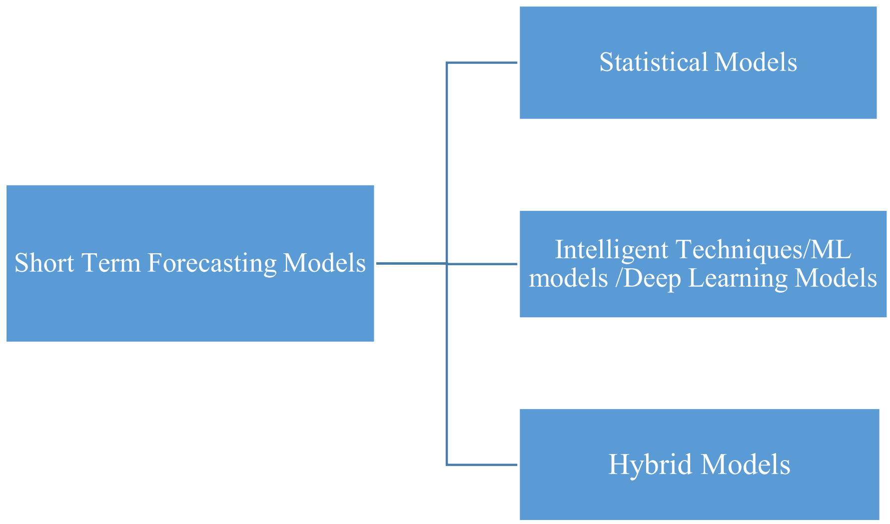


Figure 2 Forecasting models for STLF [10].

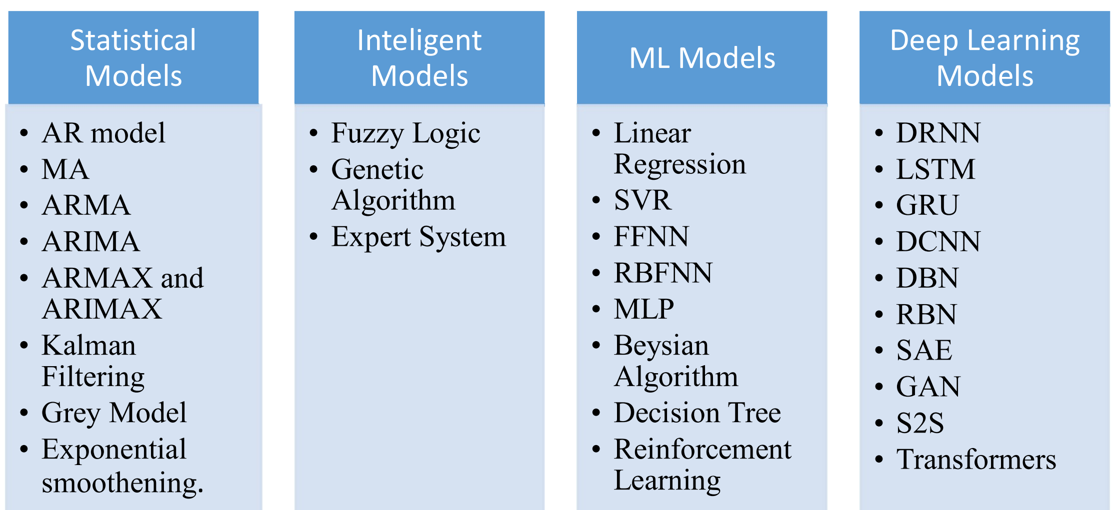


Figure 3 Models in use for forecasting [3] [4] [8] [10].

CHAPTER 2

LITERATURE REVIEW

Reference [1] uses different ML algorithms for forecasting of Solar Irradiation. The algorithms used are MLR, SVR, ANN and Random Forest. The Paper brings out the usage of these algorithms in different areas like food chemistry, economics, bioinformatics, environmental science and regression. The weather variables selected for the purpose of forecasting are latitude, longitude, altitude, average temperature, average relative humidity and precipitation. The models mentioned above has been developed using database of weather variables collected. The statistics used to analyse and compare the different algorithms that have been used are r square (squared correlation coefficient), RMSE (Root Mean Square) and Error. It has been found that ANN gives the lowest RMSE and higher 'r square' value among all the models developed. The models can be further developed using data from other meteorological sites and including more weather variables.

The application of Artificial Intelligence and Machine Learning (AI & ML) in the field of Electrical Engineering is wide [2]. It has its application in power system, Electrical Machine and Drives, Power electronics and Converters and Renewable Energy. In the field of Generation, AI & ML can help in Load Forecasting which shall help in expansion and planning. It also helps in unit commitment and economic load dispatch. Power system operation and control can be another application at the transmission level. Fault detection, relays and circuit breaker can widely use AI & ML at the distribution level. AI & ML has been used for motor conditioning, speed control, real time diagnosis, fault prediction at incipient stages, detection of inter turn faults and short circuit, steady state and transient analysis of electrical machines and vector and direct torque control of induction motor. Deep learning based algorithms can be used to develop fast and efficient controllers for power electronics converters. AI & ML has its use in power electronics which includes, non-linear function generation, delay less filtering and wave form processing, feedback signal processing of vector drive and pulse width modulation of two level and multilevel inverters. The main uses of AI & ML in the field of Non-Conventional energy sources are,

- Design of MPPT (maximum power point tracking) controller
- Design of active and reactive power controller
- Solar panel tracking
- Solar and Wind power forecasting.

Deep learning which is a subset of machine learning, has become an important tool for renewable energy forecasting. Reference [3], [4] reviews the different deep learning models for renewable energy forecasting and draws a comparative analysis. Temperature dependent forecasts can be classified into physics based models, statistics based models and neural network based models [3]. Physics based models are traditional methods which takes into account physical weather parameters and uses it in the formulae to make a forecast. Reference [3] uses sensor readings from different sites to collect data and forecasts based on that. In the statistical based methods, the forecasts are based on historical data. Neural Network based forecasting is highly efficient and yields better result. This paper has used simple recurrent neural network(SRN), Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU), Convolution Neural Network (CNN). It has also used hybrid models like GRU-LSTM parallel network and CNN-LSTM parallel network. Some of the other deep learning algorithms are stacked auto encoder (SAE), Deep Belief Network (DBN), Extreme Learning Machine (ELM) and Stacked Extreme Learning Machine (SLEM) [4]. SRN can be further modified with deep input to hidden function, with deep hidden to output function, with deep hidden to hidden function and stacked RNN [4]. The paper [4] also talks about probabilistic models which focusses on assigning a probability to each prediction result. This can be further classified into parametric and non-parametric probabilistic model. Parametric based models uses prior distribution, while non-parametric methods based models are developed using distribution free parameters.

Ensemble approaches are meta learning procedure which pools various statistical and machine learning procedure into one extrapolative model in order to minimise bias, variance and develop more accurate model [5]. There are three kinds of ensemble methods;

1. Bagging method which mainly consist of bootstrap aggregation and random forest.
2. Boosting
3. Stacking

In the ensemble methods, several different estimators are developed and finally these individual estimators are aggregated to obtain the final result. The complete data stet is broken into small samples. These trials then lay open to to regression and individual anticipated value is obtained. Final prediction is obtained by the accretion of individual prediction. This type of model helps in improving the performance of forecasting and minimizes the risk of irrelevant forecasting.

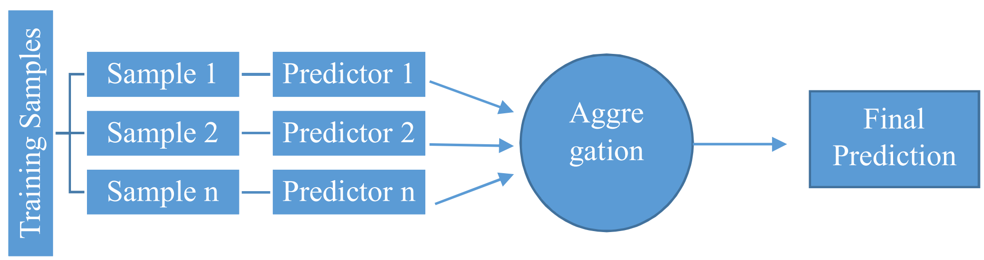


Figure 4 Block diagram of ensemble model [5].

MATLAB is a strong platform for performing neural network related tasks. It is equipped with various tools which makes it easier for users to build AI & ML models. Reference [6], [16] has used MATLAB to build ANN models for forecasting. It also allows users to write codes on text editor and develop the customised neural network model. ANN tool box in the MATLAB has inbuilt functions which help in designing, implementing, visualising and simulating the neural networks in an easy way. This dissertation uses MATLAB to develop ML models and has been discussed elaborately in chapters following.

Reference [7] proposes the use of two types of Radial Basis Function Network (RBFN)Generalised Recurrent Neural Network (GRNN) and Probabilistic Neural Network. RBFN consists of two layers, a hidden layer with non-linear neurons and an output layer with linear neurons. GRNN is an evolution over the basic architecture of RBFN. GRNN has a slightly different second layer. Number of neurons in the first layer is equal to number of input vectors and each neuron's weighted input is the distance between the input vector and its weight vector. Each neuron's net input is its weighted input with its bias. The input is then passed through the radial basis layer. The second layer also has number of neurons equal to the number of target vectors and is purely linear layer. The output of the network depends on the distance between the input vector and neuron's weight vector. Lesser the distance, steeper will be the radial basis function and the neuron with the weight vector closest to the input will have a much larger output than other neurons. Larger the distance, smoother the slope of radial basis function and several neurons can respond to an input vector.

Probabilistic Neural Network is a type of RBFN which is adapted to give values of output corresponding to the class conditional densities [7]. It is a classifier that maps input patterns in a number of class levels [7]. The network is organised into a multi-layer feed forward network with input layer, pattern layer, summation layer and the output layer [7]. PNN is an implementation of a statistical algorithm called kernel discriminant analysis [7]. The advantages of PNN are as follows: it has a faster training process and there isn't any issue of local minima [7].

A novel method for the prediction of renewable generation and load forecasting using stacked gated recurrent unit –recurrent neural network (GRU-RNN) [11]. GRU is an evolution over basic RNN and is very similar to LSTM. A GRU unit is composed of a reset gate that decides how much of the information from the previous time steps can be forgotten, an update gate that decides how much of the information from the previous time steps must be saved and a memory that brings informations along the entire sequence and represents the network of the network [11].

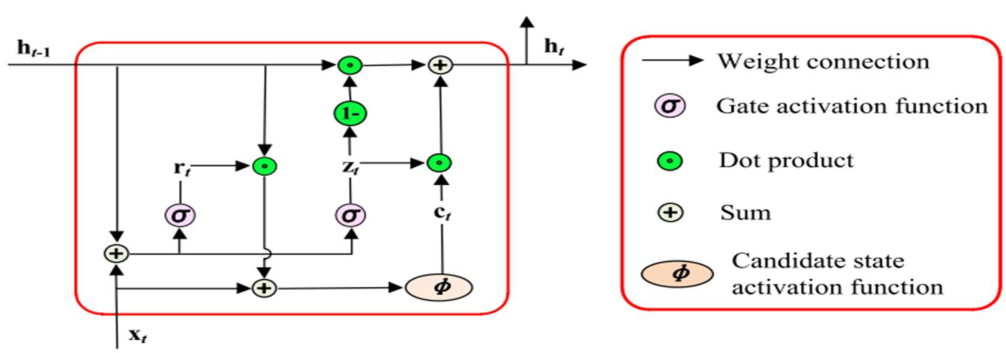


Figure 5 Model architecture of a basic GRU –RNN[11].

However, in order to get reliable and accurate plotting, GRU –RNN layers are put one after another to construct a piled GRU –RNN. This helps now negating the effect of nonlinear and nonstationary characteristics of parameters.

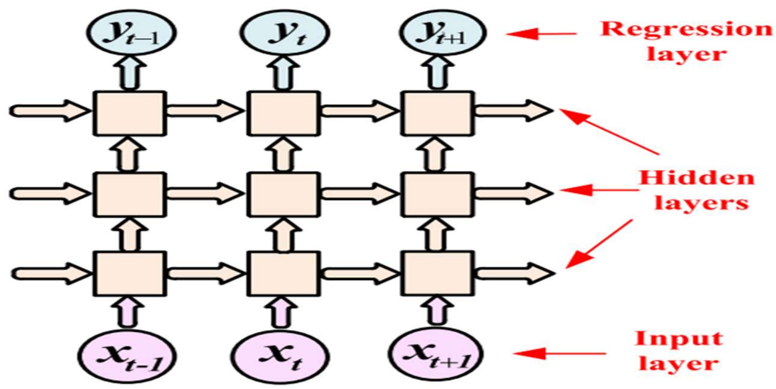


Figure 6 Stacked GRU – RNN [11]

Reference [12] presents the use of AI and ML in the field of epidemiology. The paper has used LSTM to predict the volume of COVID19 patients. This has helped the local administration in taking necessary actions to counter the spread, control hospital load and optimal allocation of resources. The author has used LSTM and have calculated the mean absolute percentage error to find the model's accuracy.

A method based on LSTM recurrent neural network has been proposed to predict the load of non-residential consumers using multiple correlated sequence information [15]. K means algorithm has also been used to analyse the daily load curves of non-residential consumers, classify and then mine the consumer's energy consumption behaviour patterns [15]. Then the spearman correlation coefficient is applied to investigate the time correlation under multiple time series for non-residential consumers [15]. It has been found that there exist multiple related time sequences such as adjacent time points, the same time points in adjacent days and the same day in the adjacent weeks among data samples for a specific consumer [15]. So, the author has proposed a non-residential load forecasting framework based on multiple sequence LSTM networks. The results have shown that this process can successfully use the multiple sequence information and successfully capture the dependencies among the sequences [15]. This proposed method can not only establish the nonlinear relationship between features and load but also capture the correlation between adjacent time point correlation, day – related correlation, and week – related correlation to improve the accuracy of load forecasting [15].

Reference [15], has a data set of 48 non-residential consumers which shows different consumption behaviour and different ways of using electricity. Hence it was appropriate to train forecasting models individually for each consumer. As these data have strong periodicity and regularity, it has been assumed that the load curves for a specific type of consumer has adjacent time point correlation, day- related and week related correlation. In order to verify the assumption k-means clustering has been used to study the pattern between different time points. The models are evaluated in terms of MAE, RMSE and MAPE.

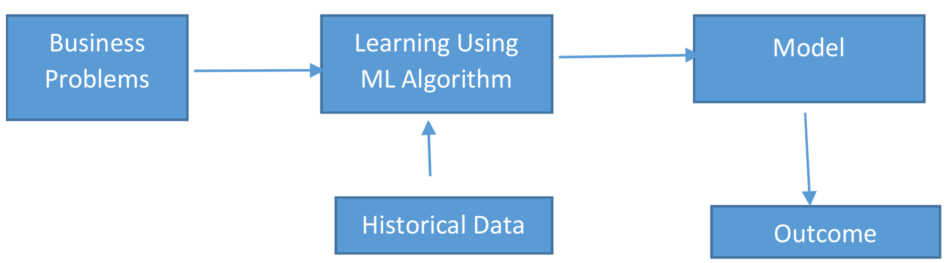
Reference [17] proposes a Robust Self Attention based Multi Horizon Model for forecasting solar irradiance. The model basically works in two steps. First part deals with multi horizon forecasting which considers multiple weather parameters. The second step is to create prediction interval and analyse it for model robustness using quantile regression. In regression we hardly measure the certainty of the predicted value because a model learns from parameters differently and gives different future values. In most of the cases, variations in the prediction will increase the length of the forecast period. In order to quantise the ambiguity of the predictions, prediction intervals are used. Hence in place of a single value, we have a range of value where we can be confident that the original values lie within it. Amount of data we have, data variation, how far the forecasting is being done and approach of forecasting effects the predictive interval. Here in this paper, quantile regression has been used to generate prediction intervals. Quantile regression is the regression technique when linear regression could not satisfy its' assumption that residuals have constant variance across values of independent variables. Quantile regression aims at estimating conditional qualities like median of the response variable. It is an extension over the linear method of regression [17] . The paper has used a self-attention based transformer model belonging to the family of deep learning models.

CHAPTER 3

BASICS OF MACHINE LEARNING AND DEEP LEARNING

3.1 Machine Learning

Machine Learning – “the study of computer algorithms that improve automatically through experience and by the use of data, part of artificial intelligence”. “Machine Learning” processes shape a model dependent on past data known as “training data” so that decisions are made without unequivocally programmed to do so. There are two main concepts in machine learning, one is training algorithm and the other is data. The training algorithm learns from the sample data and a final model is build form the learned pattern. This model is used further on the new data for regression, classification and other purposes. ML is different from traditional programming. In tradition programming business requirements are converted into deterministic rules. These rules are then applied on input data to produce outcomes. The rules are deterministic in the sense that they do not change from the data they ingest or the outcomes they produce. However, In Machine Learning we will not write deterministic rules. Instead we use ML to learn from the historical data fed and create a model, which is essentially the set of rules that algorithm has learnt from the data to solve a complex problem for writing static rules is unsuitable. Below block diagram shows the basic working of ML,



Machine Learning can be classified in the following ways:

1. Classification based on whether they are trained with labelled target values
    - Supervised Learning

The input data consists of labels or the target values. E.g. K-Nearest Neighbours, Linear Regression, Logistic regression, Support Vector Machines, Decision Trees, Random Forests.

- Unsupervised Learning

The input data does not include any label or target. E.g. k-Means, DBSCAN, PCA,

t-SNE, Apriori.

- Semi Supervised Learning

In semi –Supervised Learning, it is normally started with a supervised system gradually and then the system gradually moves into an unsupervised learning. E.G Deep Belief Networks.

- Reinforcement Learning

In Reinforcement Learning, an agent Tries to find an optimum path to solve problem based on a reward / penalty basis. The Target has to be achieved in maximum reward path.

2. Based on Whether or not they can learn incrementally on the fly
    - Online Learning

In Online Learning (also termed as incremental and out of core Learning), the model is trained incrementally by feeding it data instances sequentially, either individually or by small groups called mini-batches. Each Learning Step is faster than batch learning method and it ensures that a productionised model is kept up to date.

- Batch Learning

In Batch Learning, the model is trained using all the available data at one go. This may take more time and computing resources if the volume of input data is large.

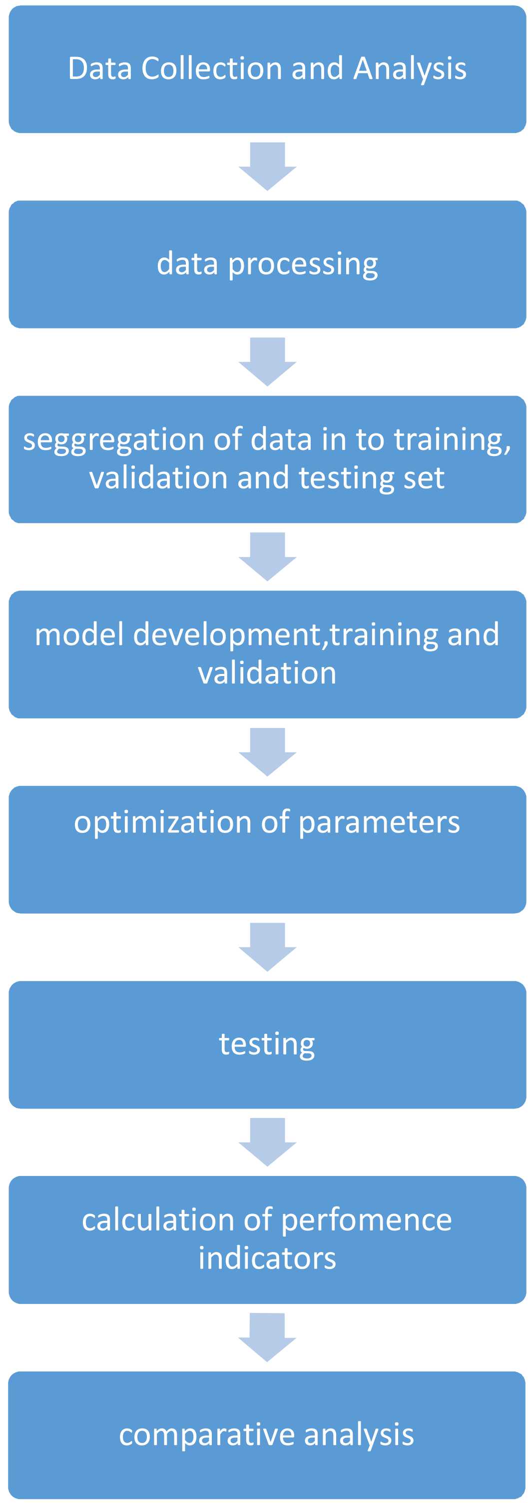
3. Whether they work by simply comparing new data points to known data points, or instead detect patterns in the training data and build a predictive model.
    - Instance based learning

The system learns from the examples fed to it and generalises it to new cases comparing them to the learned examples.

- Model based Learning

The system learns from the input examples to build a model of these examples and use to make predictions.

3.2 Workflow of Machine Learning or Deep Learning



3.3 Data in Machine Learning

Data is the heart of ML model. Learning Algorithms are applied on data to learn patterns and create models to solve a problem such as regression and classification problems. The quality of input data hence is pre requisite for the successful development and accuracy of ML model. There are different types of data that are used in developing ML models. These data can be classified into Numerical data, Categorical Data, Time Series Data, Text Data and Image Data.

It is of high importance that the raw data collected are to be pre-processed before feeding it to the relevant algorithm. Data collection is the first and the foremost step in the preparation of data. The raw data should be error free collected from a reliable source. In this work, Maximum Temperature (°C), Mean Wind Speed (m/s), Mean Relative Humidity (%), Precipitation(mm) and Solar Irradiation (kWh/m^2) are the data used to develop the model. The Source of Data has been Power Data Access Viewer which contains geospatially enabled solar related parameters for designing renewable energy systems. The data has been collected for the year 2019, 2020 and until October for 2021. The Description of the data frame which contains 1035 entries is as follows,

	Temperature	RHumidity	precipitation	windspeed	solarirradiation
count	1035.000000	1035.000000	1035.000000	1035.000000	1035.000000
mean	25.761159	49.645836	4.528522	3.572850	4.793681
std	7.788581	19.283119	8.269749	1.504201	1.601553
min	7.480000	6.810000	0.000000	0.550000	0.350000
25%	19.525000	35.280000	0.000000	2.440000	3.630000
50%	27.790000	48.560000	0.150000	3.240000	4.860000
75%	31.535000	65.470000	4.155000	4.340000	6.075000
max	40.260000	90.750000	61.740000	7.770000	8.010000

Figure 7 : Description of Data

Pearson correlation is found out between solar irradiation and all other variables. It gives the relationship, association and direction between two variables. It helps us to understand how closely two variables are related. The values of correlations are as shown below

```
solarirradiation    1.000000
Temperature         0.646454
windspeed           0.480328
RHumidity          -0.491520
precipitation      -0.527602
```

Figure 8 : Correlation Values for Different Variables

The Pearson correlation matrix shown below resembles the correlation between variables.

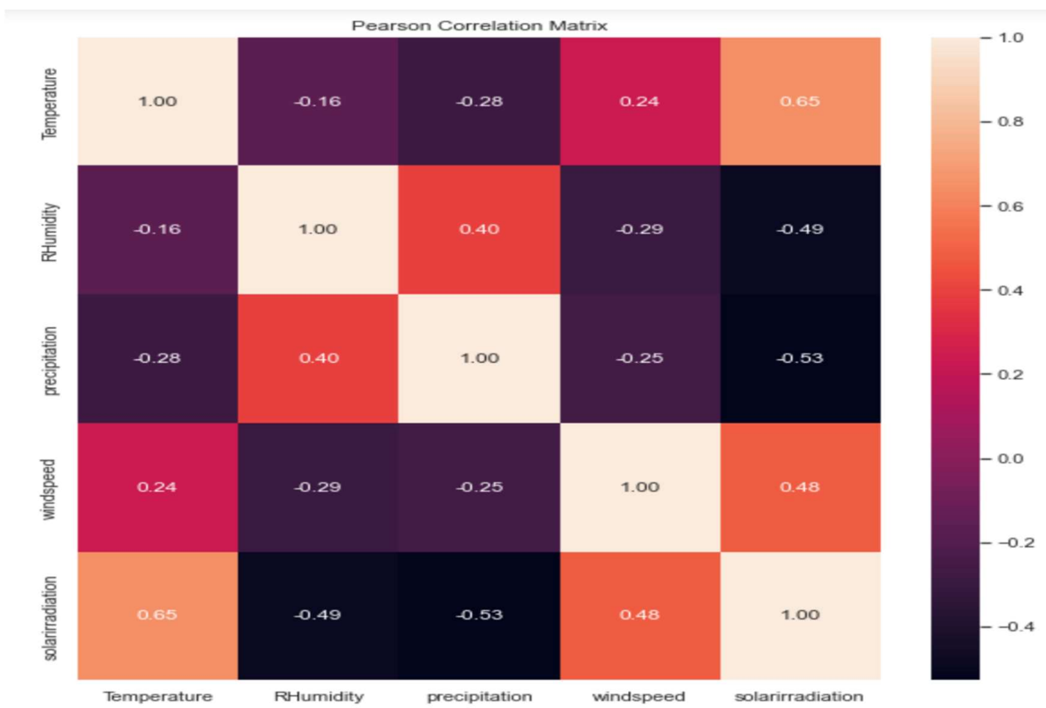


Figure 9: Correlation Matrix.

Autocorrelation is used to study the correlation of one variable at different instantaneous times. It is used to identify an appropriate time series model. Autocorrealtion is the correlation of the same series at different time instants.The correlation is calculated between two time steps called lags. Autocorrelation of solar irradiance between (t) and (t-2) instances is calculated. The autocorrelation comes out to be 0. 714. Partial auto correlation is the calculation of autocorrelation removing the intervening observations.The funtions used for calculation is acf and pacf. The plot for acf and pacf with 50 lags are shown below,

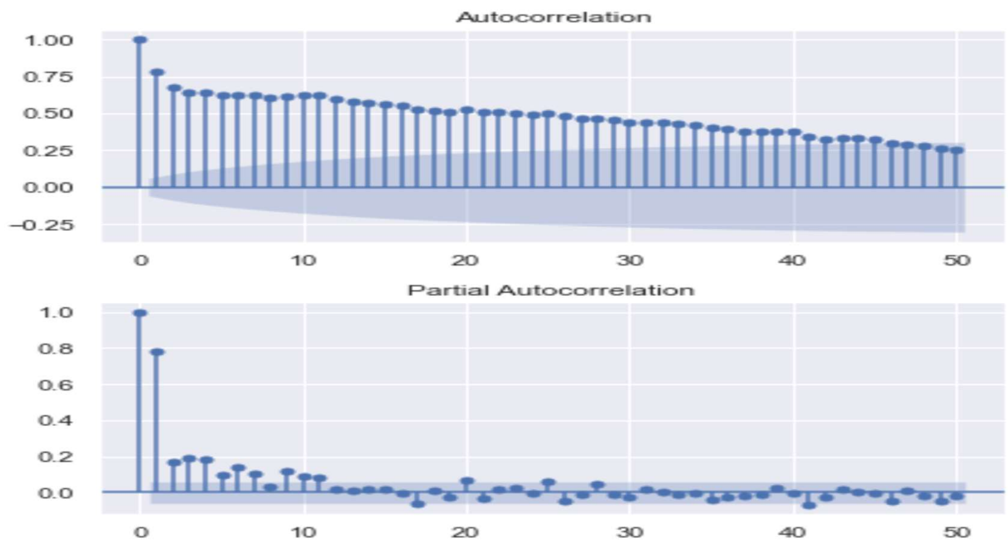


Figure 10 : Auto Correlation and Partial Auto Correlation.

Data Normalization which has been used in the thesis, is the process of transforming the features to be on a similar scale. This improves the performance and training stability of the model. There are different normalization techniques such as scaling to a range, clipping, log scaling and Z score. Z score technique has been used on MATLAB while Min Max Scaler have been used on Python.

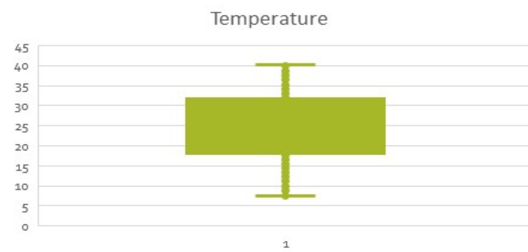
Z Score is a variation of scaling that represents the number of standard deviations away from the mean. Formula for Z score calculation is

Z score=(x-mean)/Standard Deviation.

Min Max scaler is another method used for normalisation of data set. It transforms all the features into the range of 0 and 1.

X norm= (X-X min)/ (X max-X min)

The following figures below show the range of raw data and normalised data for temperature.



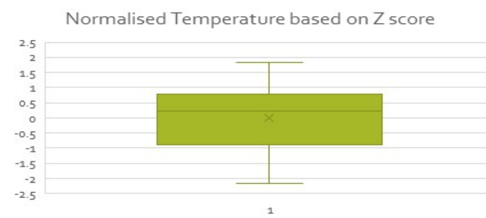


Figure 11: Raw Temperature

Figure 12: Normalised Temperature.

3.4 Activation Functions

the neural networks or deep learning algorithms contain an important section called activation function. It is a mathematical function that get the input values into a particular range. As for example a neural network is fed with inputs that are real numbers and its weight matrix is initialised randomly. It is needed for classification that is the output value is needed to be between zero and one. When the output values are not within the expected and specific range, the activation function scales the output to a specific range.

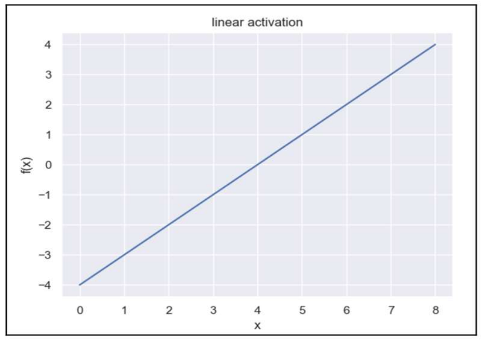
$$A=f\left(\sum_{i=0}^{n} \mathrm{Wi}.\,\mathrm{Pi} + \mathrm{B}\right), \text{ f is the required activation function.}$$

There are numerous activation functions based on the need. The examples of some of the activation functions are linear, sigmoid, tanh, ReLU, softmax, etc. Here some of the activation functions that are used in the thesis have been discussed.

Linear Activation

This function directly changes with the input. It scales the output by an appropriate factor. The below shown is the plot of f(x) with respect to input x for linear activation,

$$f(x)=cx$$

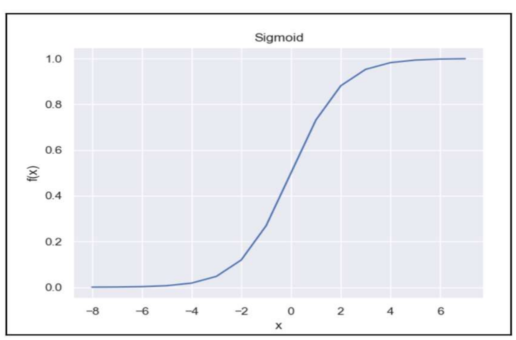


Sigmoid activation

For all inputs that are real, the output varies between zero to one. It is necessary for generating probabilistic scores from the build biological neurons. It helps to retain the non-linearity of outputs as the function is continuous and S shaped. The slope of the curve is higher initially near the axis and saturates as we go away from the axis. The shape is similar to the magnetisation curve. This means that for even a small change in input there will be reasonable change near the origin and there won't be significant changes at the saturation. This quality helps in the segregation work as it keeps the output near to absolute values. The equation for sigmoid activation for the input x is given by,

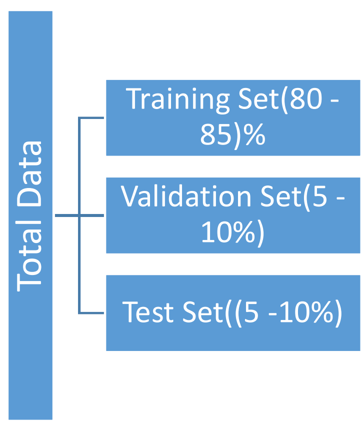
$$f(x) = \frac{1}{1+e^{-x}}$$

The figure shown below is the curve of the sigmoid activation function,



3.5 Model Development

Development of model is a combination of steps in a sequence. Usually the first step is to split the data set into training, validation and testing set. We have to go for splitting of data in order to verify the working performance of our trained model and check in which direction the training of the model is moving. The training set contains the data that will be used to train the model. This set contains the majority of the data and is largest of all. The model learns the features and patterns from the training set by making iterations. Even though accuracy of the training is calculated, it doesn't really make the evaluation of the model. High accuracy during training may not give us an accurate picture of an entire model. The validation set provides frequent and unbiased evaluation for the model. The Validation set aids in fine tuning the parameters during training. The test set is generally reserved from the total available data that the model hasn't seen during training or validation. The test set allows to evaluate our model with unknown data. Generally, it is good to split the available data in the ratio 8:1:1.



Training the network implies learning the optimal weight values starting from the initialised set of weight values. This make it possible to find a perfect set of weight values that will give us perfect prediction results. In order to go from initialised weight values to optimal weight values, we need to go through these three steps:

- Loss function helps us to find out how much does the forecasted output varies from the output that we want. It informs if the initial values are best fit.

- Maximum likelihood/occurrence estimation(MLE) is the next step ahead, which retaliates that in order to get an optimized set of parameters, the probability of achieving the expected values is to be maximised. In a better way to understand, the target should be minimizing the loss function by changing the weights.

- The above two points leads to the conclusion that the loss function is to be as low as possible by improving the weights continuously. It can be said that the network has understood the portraying function to get values close to anticipated values, when the minimum possible value for loss function is reached.

Loss Function

The loss function is the degree of correctness of prediction and forecasting. In simple terms loss function is defined in terms of the mathematical difference between the anticipated and the real predicted outputs shown as L(w):

L(w)=y(expected)-y(predicted).

However, this simplest form of loss function is error prone and various other loss functions have been used for regression tasks. The most common regression loss that is in use is mean squared error. This is defined by the following equation:

$$MSE = \frac{\sum_{i=1}^{n}(y_i - y)\ \hat{}2}{n}$$

Where ‘y’ is the value that is predicted and $'yi'$ is the corresponding expected value. The parameter ‘n’ is the amount of data samples. As evident from the name, the error function at first takes the difference between predicted and expected values (error) and then squares it and finally divides it by the quantity of samples. This gives the average value of squared error. The error is squared before summing to make all values on the right of zero on scale. This way we are only looking at the effect of magnitude of the error. Further, mean is calculated to keep the loss standardized and evade unforeseen great loss values. Due to the squared values, any data far away from likely value will give much more to the loss than it should have.

Optimizers play an important role in minimising the cost function by varying the learning rate continuously at the time of training. Optimisers are used to get better convergence of output. Some of the optimisers that have been used on MATLAB are Scaled Conjugate Gradient Algorithm (SCG), Levenberg Marquardt Algorithm (LM) and Bayesian regularisation algorithm (BR). Most used optimisers on Tensor Flow are Adam optimizer and RMSProp.

LM algorithm is the most widely used optimization algorithm which is a mixture of "vanilla gradient descent" and "gauss-newton iteration". "Vanilla gradient descent" is the easiest method to find lowest value of a function. To Keep the values of parameter up to date, addition of the opposite of the "scaled gradient" is needed,

$$X(i+1) = x\,(i) - \lambda\,(\nabla f) \ldots\ldots. (1)$$

"Simple gradient descent" undergoes convergence issue. It should take large steps down the gradient at locations where the gradient is small and vice-versa. However, it does just the opposite. Another issue is that the curvature of the error surface may not be same in all directions. These situations can be improved by curvature as well as knowledge of slope, namely "second derivatives". One way to do this is to use "Newton's "method to solve the equation $\nabla f\,(x) = 0$. Expanding the gradient of 'f' using "Taylor series" around the current state x^0, we get

$$\nabla \mathrm{f(x)} = \nabla \mathrm{f}\,(x^0) + (x - x_0)^T\,\nabla^2\,\mathrm{f}\,(x^0) + \text{higher order terms of } (x - x^0) \ldots\ldots. (2)$$

Neglecting the higher order terms (assuming f to be quadratic around x^0), and solve for the minimum x by setting the left hand side of (2) to 0, we get the updated rule for Newton's method –

$$x(i+1) = x(i\,) - (\nabla^2 f(xi))^{-1}\ \nabla f(xi) \ldots\ldots. (3)$$

where x^0 has been substituted by $x(i)$ and x by $x(i+1)$.Since newton's method unconditionally uses a quadratic assumption on 'f', this technique has an advantage of rapid convergence. It can be hence seen that simple gradient descent and newton's method is complementary to each other. Based on this the LM algorithm makes updating. If the error has increased as a result of update, then the steps are retracted to reset the weights to previous values. If the error has decreased, then the weights are kept at new values.

BR Algorithm which uses Bayes' theorem for regularisation. In doing so it penalises models based on their complexity favouring simpler models that ae also better at generalizing. "SCG" according to the scaled conjugate gradient method bring up-to-date weight and bias values. It provides faster training and excellent test efficiency because it is a second order training algorithm for training of neural network. It is the subclass of Conjugate Gradient Methods, which has the ability to converge on most of the time. SCG avoids a time consuming line search per learning iterations by using a step size scaling mechanism, which makes the procedure quicker than other second order algorithms.

Some of the optimisers that have been used in deep learning are RMS Prop (Root Mean Square Propagation) and Adam (Adaptive Moment Optimization). The essence of "RMS Prop" is that it upholds a discounted average of the square of gradients and then divides the gradient by the root of this average. This operation of RMS prop uses plain push. "Adam optimization" is a "stochastic gradient descent" method that is based on adaptive estimation of first order and second order moments. This method has little memory requirement, computationally efficient, invariant to diagonal rescaling of gradients, and is well suited for problems that are large in terms of data /parameters.

3.6 Model Evaluation

Once the model is developed, it is tested by feeding test data set or a different data set that the model has not seen. This thesis while working on MATLB uses a new data set which is used to predict the solar irradiance of 10 months from January to October for the year 2021. The data The data contains the input variables similar to the input of trained neural network. The number of samples that are present is 304.Predicted solar Irradiance is compared with actual solar irradiance for the year 2021 and is plotted with respect to the number of samples. While working on Python testing data set is used to evaluate the model. The predicted solar irradiation and actual solar irradiation for the year 2021 with respect to number of test samples is plotted.

3.7 Key Performance Indicators

Performance metrics are the part of every machine learning pipeline. They tell us about the progress of the model and give us an opportunity to judge the performance. The thesis uses regression models which have continuous output. The metrics used for performance measurement are discussed below,

- Mean Square Error

It is the record prevalent method for regression difficulties. It finds the average of the squared difference between the actual value and the value predicted by the regression model.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(yactual - ypredicted)^2$$

MSE can be optimized better because it is differentiable. It squares even small errors, which principally tells how bas the model is. it’s fundamentally, more prone to outliers than other metrics.

- Mean Absolute Error

MAE is the average of the difference between the truth and the predicted values. it can be epitomized as

$$MAE = \frac{1}{N}\sum_{j=1}^{N}|yactual - ypreicted|$$

It is extra vigorous towards outliers as it doesn’t overstress errors. It measures of how far the predicted value were from the actual output. However, it doesn’t give us an idea of the direction of the error, i.e. whether we’re under-predicting or over-predicting the data. Error interpretation needs no second thoughts as it perfectly aligns with the original degree of the variable. It is a simple metrics to implement.

- Root Mean Square Error (RMSE)

"Root Mean Squared Error" is equal to the square root of the mean of the squared difference between the actual value and the value predicted. Basically it is sqrt(MSE). It can be epitomized as,

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(yactual - ypredicted)^2}$$

It addresses a few downsides in MSE. It keeps in mind the differentiable property of MSE and knobs penalisation of minor errors done by MSE by taking the square root of it. It is less prone to outliers as the scale factors are normalised.

- Mean Absolute Percentage Error (MAPE)

MAPE is one of the most prevalent way of calculating the accurateness. It is suggested due to its' advantage of scale –independency and interoperability. It is the average of absolute percentage errors.

$$MAPE = \frac{1}{N}\sum_{j=1}^{N}\left|\frac{yactual - ypreicted}{yactual}\right|$$

- R^2 Coefficient of determination

It in fact is a metrics calculated based on other metrics. This metrics gives us the information about the changes in target with respect to the deviation in regression line. It is the proportion of variation in the dependent variable that is predictable from the independent variables. it is expressed as,

$$R^2 = 1 - \frac{SSresidual}{SStotal}$$

SSres= residual Sum of Squares=$\sum_i(yactual - ypredicted)^2$

SStotal = total Sum of Squares=$\sum_i(yactual - ymean)^2$

when the modelled values exactly match the observed values, results in SSres to zero and R^2 to 1.

CHAPTER 4
FORECASTING USING ML ON MATLAB

MATLAB is defined as “programming and numeric computing platform used by millions of engineers and academicians to analyse data, develop algorithms and create models”. It combines a programming language that expresses matrix and array mathematics directly with desktop environment tuned for iterative analysis and design process. It has inbuilt live editor for creating scripts that combine code, output, and formatted text in a notebook. It is a professionally built platform that has applications developed for specific purposes. These MATLAB apps allows the users to see how different algorithms work with the available data. This can be iterated until the desired results are obtained and then automatically generate a MATLAB program to reproduce or automate the work.

Using MATLAB, thousands of machine learning algorithms have been developed. MATLAB makes working with machine learning easy with,

- Point and click apps for training and comparing models
- Advanced Signal Processing and feature extraction techniques
- Feature selection, model selection and parameter tuning
- All popular classification, regression, and clustering algorithms for supervised and unsupervised learning.
- Faster execution

MATLAB contains interactive tools and applications for machine learning algorithms alongside other deep learning algorithms. These can be used for classification, regression and clustering purposes. These applications and tools can train, compare, tune and export models for further analysis, integration and deployment. MATLAB also contains live editor where customised codes can be written for developing machine learning models. This give the user an opportunity to change parameters according to their need.

4.1 Multivariate Linear Regression (MLR)

MLR is a conventional machine learning algorithm which relates multiple predictor variables to the target variable. It studies the cause effect relationship between dependent and independent variables. In MLR the relationship between each independent variable and the dependent variable is examined. The basic equation for MLR is shown below,

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 \ldots \ldots \ldots$$

Where y is the dependent or the target variable, ($X_1, X_2, X_3, \ldots$) are the predictor or the independent variable, ($\beta_0, \beta_1, \beta_2, \beta_3 \ldots$) are the regression coefficients.

The functional connection between the dependent and independent variable can be stated in the form of a matrix. A matrix of variable multi linear of p regression cab be written as below,

$$\begin{bmatrix} Y_1 \\ Y_2 \\ . \\ . \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & x11 & . & x1p \\ 1 & x21 & . & x2p \\ . & . & . & . \\ 1 & xn1 & . & xnp \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ . \\ . \\ \beta_n \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ . \\ . \\ e_n \end{bmatrix}$$

where 'Y' is an output variable vector of size $n \times 1$; 'X' is an input variable matrix of size $n \times (p + 1)$; 'b' is a coefficient vector of size (p + 1) ×1 and 'e' is an error vector of size $n \times 1$. The regression coefficients can be found out with least square methods.

In this thesis, solar irradiation is taken as the target variable that is to be predicted using independent variables which are temperature, relative humidity, precipitation and wind speed. For developing the MLR model, the output i.e. solar irradiation is written in terms of weighted sum of all the input variables. The coefficients of the input variables are found out using the 'regress' function. The coefficients, which is a column matrix, are used to calculate the output based on the data for the year 2019 and 2020 as follows,

$$Y\ predicted = b\ (1,1) + b(2,1).* x1 + b(3,1).* x2 + b(4,1).* x3 + b\ (5,1). * x4$$

The linear model developed above is used to predict the output for the next 304 days for the year 2021.

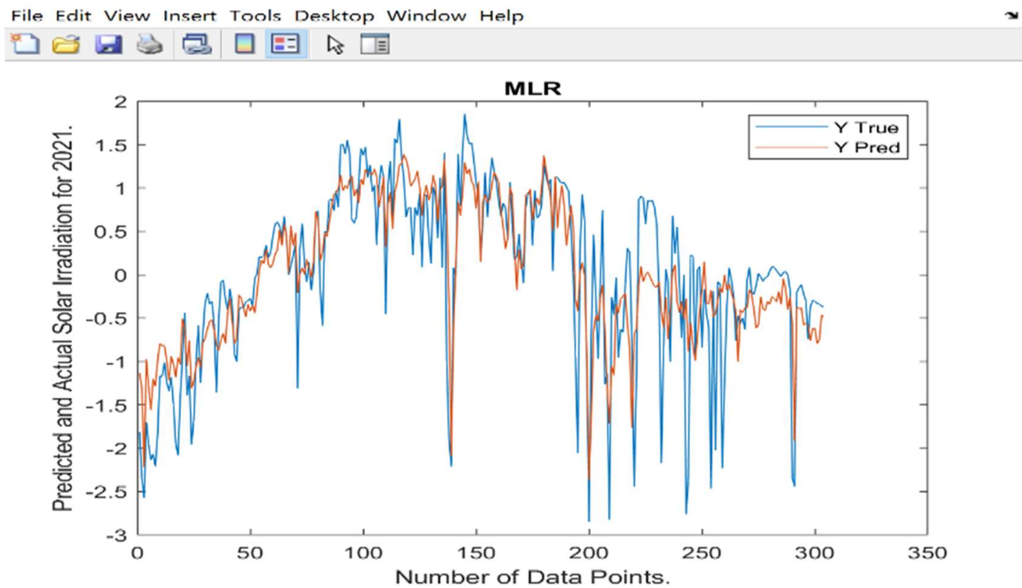


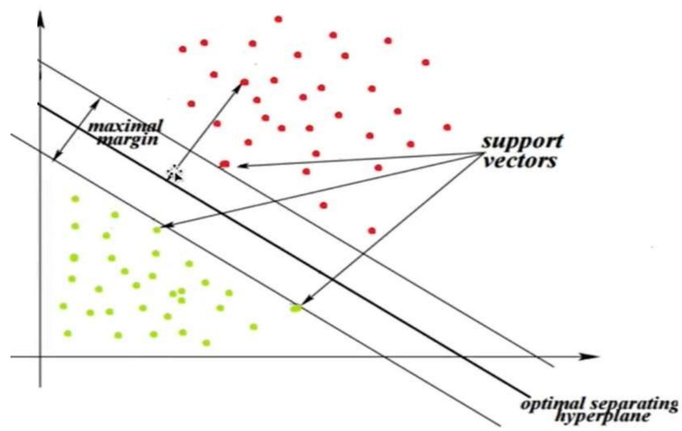
Figure 13 Prediction of Solar Irradiation Using MLR

The above figure shows the predicted and the actual solar irradiation for the year 2021. The values of the performance indicators calculated is shown below,

MSE	2.47 e^{-4}
RMSE	0.0155
MAPE	0.4044
MAE	2.381
R Square Value	0.6846

4.2 Support Vector Regression (SVR)

A "support vector Machine" (SVM) is a potent and multipurpose Machine Learning model, capable of performing linear and nonlinear classification and regression. It is one of the most general models in machine learning and is mostly suitable for classification of complex data sets that are small and medium sized.



SVM can be explained with the help of a figure above. The data has two classes as noted by the colours and they are clearly linearly separable. The goal of the SVM is here to find the widest possible street between the classes. This is called is the large margin classification. The bold line in the middle is the decision boundary. The instances located on the edge of the margin is called support vectors. So the objective of SVMs is to infer a precise decision rule with a suitable generalisation ability by picking some specific subset of training data, called "support vectors".

"SVM" which have been primarily developed for classification problems, however it has been widely used for regression problems. It is usually used with classes which are binary in nature even if they are non-linearly separable in 2D space, and this is accomplished by bring together a trick that changes the binary classes into a higher dimensional space where the classes become linearly separable, and this trick is called "kernel trick". The basic kernels are polynomial radial basis function and linear. So some of the properties of the SVM are inherited by SVR. In SVR, it is basically a classification of the regression errors that are greater or less than a certain edge value instead of the classification being between the two appropriate sections.

In this thesis, a SVR regression model is developed on MATLAB to predict solar irradiance. The function that has been used to train the model is 'fitrsvm'.'fitrsvm' trains or cross validates a support vector machine regression model on a low to moderate dimensional predictor data set. The model is optimized for hyper parameters. The optimization takes 30 iterations and time take taken is around 757 seconds. The appropriate values of parameters like box constraint (0.00684), epsilon (0.13066), kernel function(polynomial), order (3) are taken to develop the model. Box constraint is a positive numeric value that controls the penalty imposed on observations that lie outside the epsilon margin and helps to prevent overfitting. The model developed is used to predict the output for the 304 days of the year 2021.

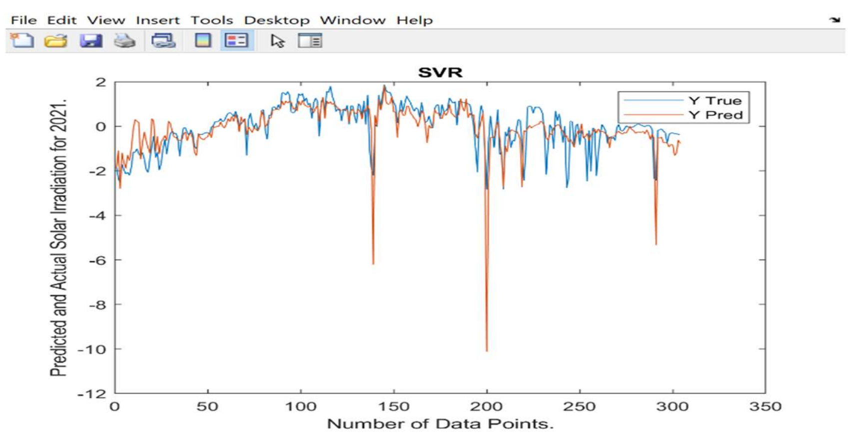


Figure 14: Prediction of Solar Irradiance using SVR

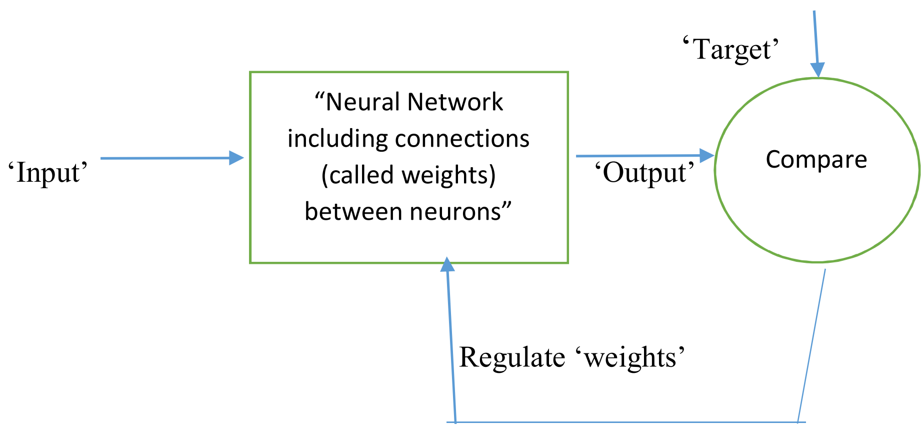
The above figure shows the predicted and the actual solar irradiation for the year 2021. The values of the performance indicators calculated is shown below,

MSE	0.0072
RMSE	0.0848
MAPE	0.4789
MAE	2.3413
R Square Value	0.3830

4.3 Feed Forward Neural Network(FFNN)

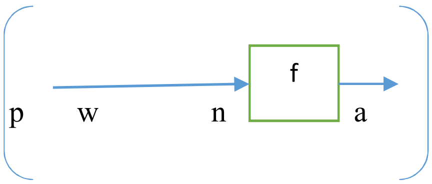
"Neural networks" contain simple elements functioning in parallel. These elements are encouraged by human nervous systems. The network function is found largely by the interconnections amid fundamentals. By adjusting the values of the connections (weights) between elements, we can train a neural network to perform a particular function.

Usually neural networks are tuned, or trained, so that we get desired output by using specific input. Block diagram of a simple situation is shown below. Here, the network is tuned, based on a comparison done between output and the target, until the target is matched by the network. In reality, many such input/target pairs are used, in this kind of supervised learning, to make a network learn.

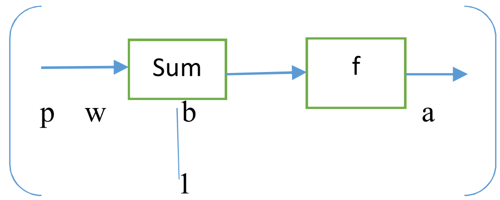


Neural networks have been trained to perform complex functions in various fields, including pattern recognition, identification, classification, speech, vision, and control systems. Neural networks are also trained to solve complex functions in various fields that are tough for orthodox computers or human beings. The field of neural networks has a history of decades but has found rock-hard use in the past fifteen years and is still evolving hastily. In recent times it has been widely used in other fields of engineering as well.

A simple neuron with a single scalar input with bias and no bias is shown below.



$$a = f(wp)$$



$$a = f(wp + b)$$

The input 'p' which is a scalar quantity is communicated through a linking that multiplies its strength by the scalar weight w, to form the product $(wp,)$, which is also a scalar quantity. The weighted input $(wp,)$ is the only argument of the transfer function f, which produces the scalar output a. The diagram of neuron on the right has a scalar bias, b. The bias is simply being added to the product wp as shown by the summing junction or as shifting the function f to the left by an amount b. The value of bias is similar any weight. It has a constant input of 1,if no value is assigned.

Net input of the transfer function on the right side is the sum of the weighted input wp and the bias b. The transfer function f takes the sum as its argument. The transfer function, may be a step function or a sigmoid function, which takes the argument and produces the output. In previous sections, several transfer functions have been deliberated. It is to be noted that w and b are both changeable scalar parameters of the neuron. The central idea of neural networks is that such parameters can be tuned so that the network exhibits some desired or interesting behaviour. Thus, we can train the network to do a particular job by tuning the weight or bias parameters. The network may itself adjust these parameters to achieve the anticipated end.

A neural network seldom has a single element as an input. The input is fed to the neural network as a vector. The specific component inputs are multiplied by weights. These weighted values are served to the summing junction. There sum is simply Wp, the dot product of the (single row) matrix W and the vector p. The neuron has a bias as stated above, which is summed with the weighted inputs to form the net input. The transfer function f the sum as the argument.

$$n = W * p + b$$

The neural networks are connected in layers. A layer of a network comprises the grouping of weights, the multiplication and summing operation, the bias b, the transfer function f. The array of inputs is not counted in in a layer. A layer may contain two or more neurons and a particular network may contain one or more such layers.

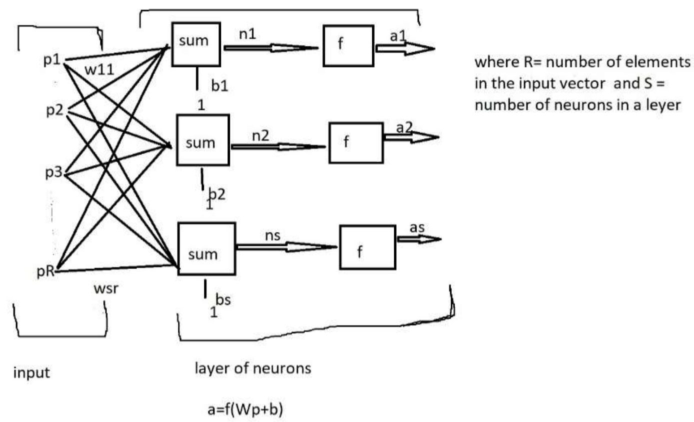


Figure 15 A single layer neural network architecture

In Figure 15, each element of the input vector p is linked to each neuron input through the weight matrix W. The i^{th} neuron has an adder that collects its weighted inputs and bias to form its own output $n(i)$. The numerous $n(i)$ taken together form a $'S'$ element net input vector n. To finish, the neuron layer yields form a column vector $'a'$. At the bottom of the figure ,the expression for $'a'$ is shown. A network can have several layers. Each layer has a Weight Matrix W, a bias vector b and an output vector a. The layers of a multilayer network play different roles. A layer that produces output is called an output layer. All other layers are called hidden layers. Multilayer networks are powerful and can be trained to approximate any function as it gives the network greater freedom. For example, any reasonable function can be represented with a two-layer network: a sigmoid layer feeding a linear output layer. The architecture of the neuron depends on number of layers a network has, the number of neurons in each layer, each layer's transfer function and how the layers connect to each other. These parameters depend on the type of problem to be solved and is up to the designer.

For developing a feed forward neural network on MATLAB, ‘feedforwardnet’ function has been used. The function takes hidden layer size and training function as the input. The other parameters like transfer function, training ratio, validation ratio are also provided while developing the model. The 731 samples have been randomly divided into Training, Validation.85% of the samples i.e. 621 samples have been taken as training data which are used to train the network which makes the required adjustment according to its error.15 % of the data i.e. 110 samples have been used as validation data. Validation data is used to measure network generalization and to halt training when generalization stops improving. The neural network architecture contains 4 inputs which feed to the hidden layers. The network uses three-layer feed forward network with sigmoid transfer function in the hidden layers and linear transfer function in the output layer. Once the network is optimised and after multiple training it has been found that two hidden layers with 20 and 10 neurons respectively gives appropriate result. The network has been trained by taking levenberg-Marquardt algorithm(LM), Scaled conjugate gradient algorithm (SCG) and Bayesian Regularisation algorithm(BR). On multiple training and optimization, BR algorithm has been taken as a training function. The feed forward network is shown below,

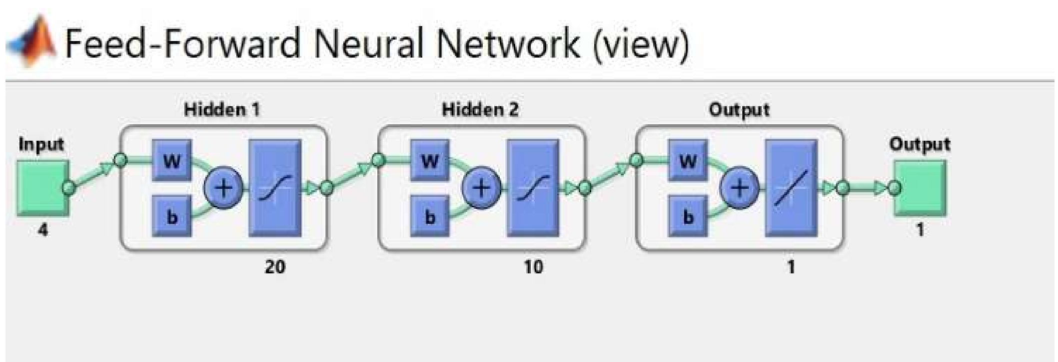


Figure 16: Feed Forward Neural Network.

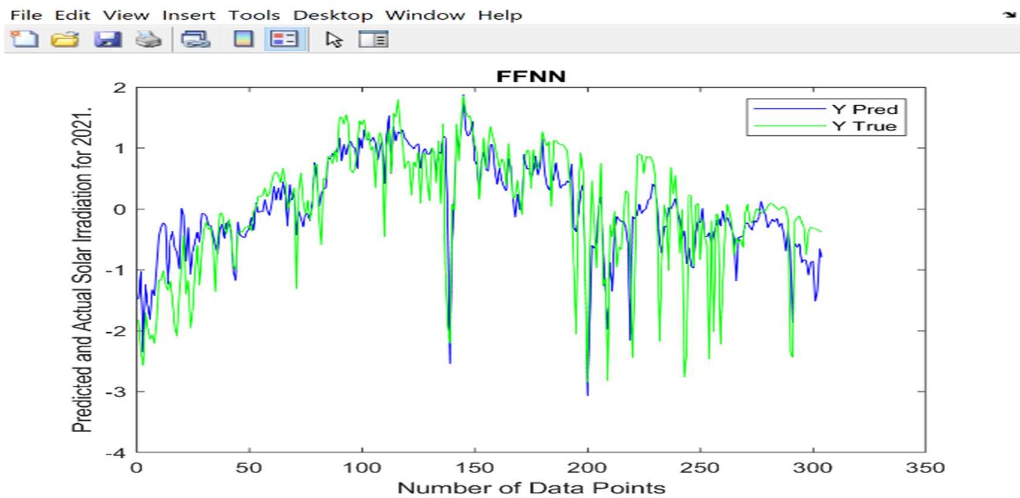


Figure 17: Prediction of Solar Irradiation for 2021 using Feed Forward Neural Network.

The above figure shows the predicted and the actual solar irradiation for the year 2021. The values of the performance indicators calculated is shown below,

MSE	2.786 e^{-6}
RMSE	0.0017
MAPE	0.4436
MAE	2.5416
R Square Value	0.6244

4.4 Layered Recurrent Neural Network (L RNN)

Layered neural network is a modification over feed forward neural network as each layer of the network has a recurrent connection with a tap associated with it. This allows the network to have an infinite dynamic response time series input data. A L RNN is a combination of 1D RNNs and is able to learn contextual information adaptively.

In order to develop layered recurrent network model on MATLAB, 'layrecnet' function has been used. The function takes three arguments as input. These arguments are layer delays, hidden size, training function. There are two hidden layers containing 10 neurons each and has been trained using Bayesian regularisation algorithm. The layered recurrent network is shown below,

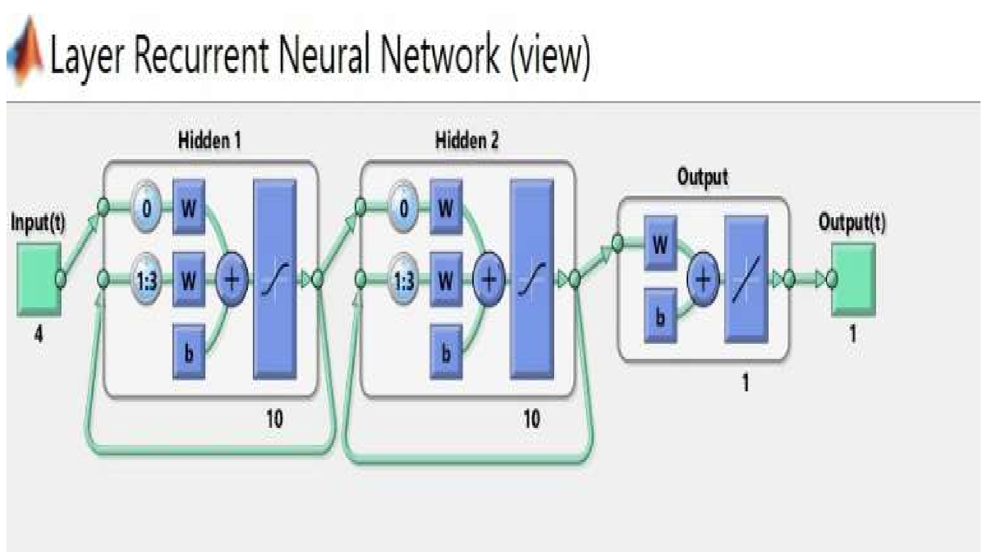


Figure 18 Architecture of layer recurrent neural network.

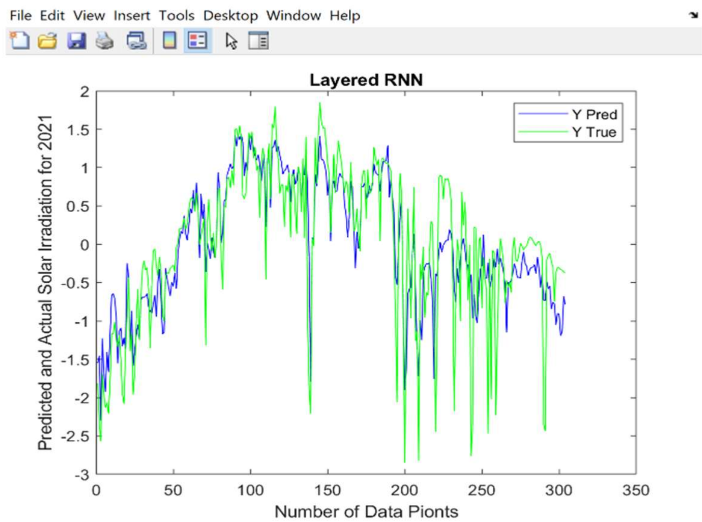


Figure 19: Prediction of Solar Irradiation for the year 2021 using LRNN.

The above figure shows the predicted and the actual solar irradiation for the year 2021. The values of the performance indicators calculated is shown below,

MSE	4.29 e^{-7}
RMSE	6.54 e^{-4}
MAPE	0.406
MAE	2.82
R Square Value	0.69

CHAPTER 5

FORECASTING USING DEEP LEARNING

Artificial intelligence is a broad set of approaches and approaches towards the direction of enabling machines to achieve human intelligence. Machine learning is a collection of techniques involving training algorithm on data to create models (rules) to perform automated tasks. Deep learning is a subset of machine learning approaches involving creation and stacking of artificial perfections in the same manner as neurons in human brains, in the form of layers. Deep learning has been inspired by human brain. In human brain there are millions of interconnected neurons which makes it possible to learn and act faster. The neurons in the mind has two constituents- dendrite and axon. The dendrite works as receptor and associates all the signals that the neuron is fetching. The axon is linked to dendrites at the end of other neurons through synapses. When the arriving sign crosses the edge, they tide through the axon and synapse to permit the signal to the linked neuron. The construction in which the neurons are linked to each other picks the network's competences.

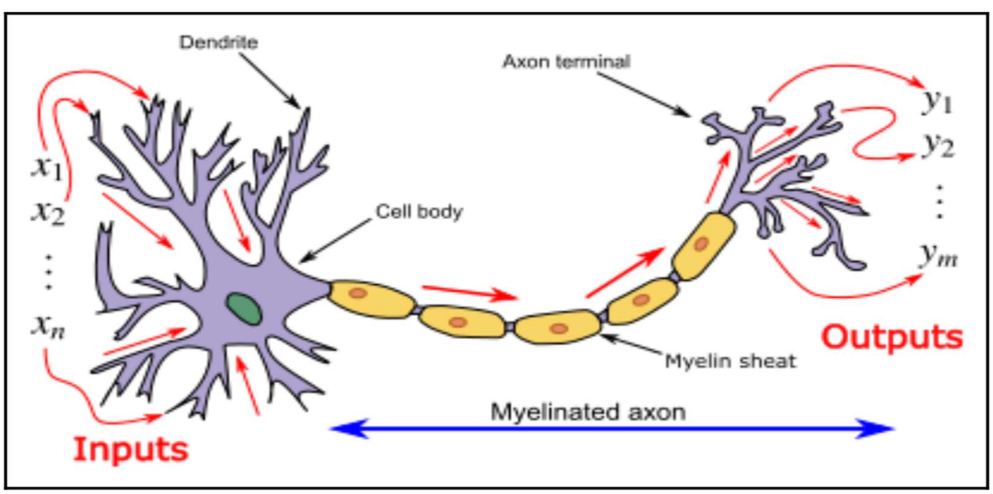


Figure 20: A biological neuron.

Deep Learning Algorithms which are widely in use for different purposes are Deep Feedforward Network, Recurrent Neural Network, Convolutional Neural Network, Auto encoders, Restricted Boltzmann Machines, General Adversarial Networks and Hybrid Networks. Deep Learning has been widely used in Industries in the field of speech recognition & classification, speech to text conversion, natural language processing and machine Translation. Deep Learning Consists of Layers, Network Architecture, Loss Function, Optimizer and an Objective function. Basic Architecture of Deep Neural Network is shown below,

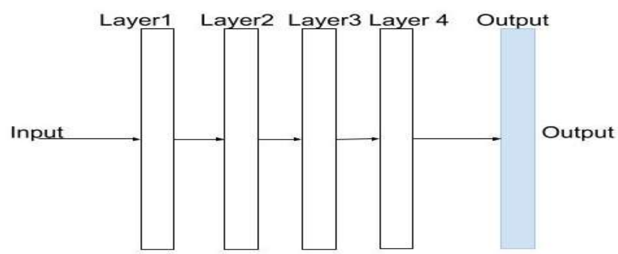


Figure 21 Basic structure of deep neural network.

Statistical Methods and shallow Machine Learning (ML) has been widely used in forecasting of temperature, solar irradiation, price forecasting. However, these traditional ML methods have limitations. Traditional ML methods have been effected by missing values, are unable to work on complex values and are usually good for short term forecasting and not for long term. This thesis proposes the application of LSTM and Sequence to Sequence Learning in Solar Irradiation Forecasting taking weather parameters as the input. The algorithms are developed on Jupyter notebook using Keras and Tensor flow libraries.

5.1 Long Short Term Memory (LSTM)

LSTM is an evolution over Recurrent Neural Network (RNN). Hence it is imperative to discuss about RNN before going to LSTM. Most of the algorithms discussed above have unidirectional flow of data. However, Deep learning models allow data to proceed in any direction. The data may also follow a feedback path that is data encompassing back from the earlier yield turn out to be a portion of the following input data. RNNs are the pronounced example data looping. The drawing shown below is portrayal of basic RNN,

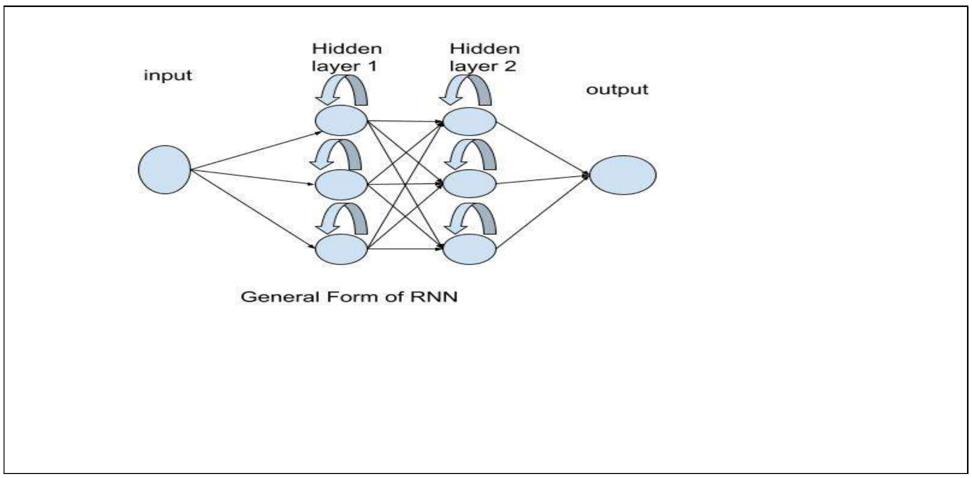


Figure 112 : General Form of RNN

From the above drawing it is seen that, information from the earlier stint points goes into the teaching of the current time point. The recurring style makes the model graft fine with time series or sequential inputs. Suppose there are some inputs Xt, where it means inputs at different time steps. In a traditional neural network, the relation between the input and the output is as follows,

$$ht = f(Xt)$$

However, in RNN the feedback loop passes information of the present state to the next. The output of RNN network at any time can be written as

$$ht = f(h(t-1), X(t))$$

This chain like structure or the memory attribute of RNN has enabled it to be used in various fields.RNN which forms the basic unit for time series forecasting suffers from the vanishing gradient problem. In vanishing gradient problem, the gradient of the loss function approaches to zero making it difficult to train the model. LSTM is an evolution over conventional RNN and helps in overcoming the limitations. LSTM contains a recollection unit and three gates are supplementary to grip long term reliance. Below shown is the recurring cubicle of LSTM,

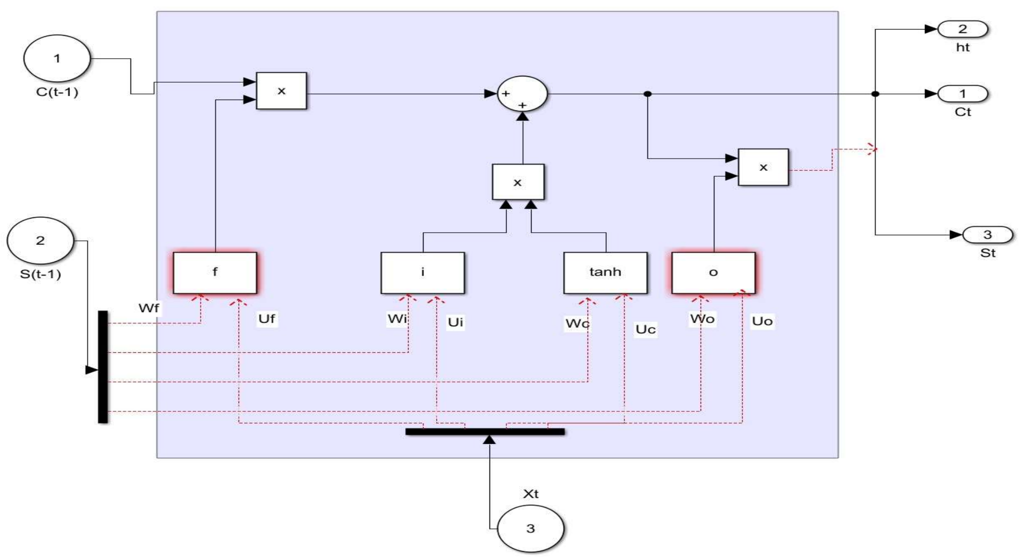


Figure 123: Recurrent cell of an LSTM RNN

The key components of the LSTM cell is explained below,

- Ct is the recollection/memory component that recapitulates the situation from the very start of the input order.
- 'f' is the forget state that wheels the flow facts from the earlier situation C(t-1). It has weights Wf and Uf connected to the hidden state S(t-1) and current input Xt respectively.
- 'I' is the input state which controls the movement of information from the existing input and weights Wi plus Ui attach it to the unseen state and the current state respectively.
- 'tanh' is computed based on the current input and hidden state connected with the weights Wc and Uc respectively.

- 'o' functions as the output and it controls how much facts from the in-house memory is used as the output of the entire recurrent cell. Wo and Uo are the respective related weights.

So, the association amid these mechanisms is summarised,

- Output of the forget gate f at time step t is computed as
  f=sigmoid (Uf xt+ Wf St-1)
- Output of the input gate i at time step t is computed as
  i=sigmoid (Uc xt+ Wc St-1)
- Output of the tanh activation c' at time step t is computed as
  c'=tanh (Ui xt+ Wi St-1)
- Output of the output gate o at time step t is computed as
  o=sigmoid (Uo xt+ Wo St-1)
- Memory unit Ct is updated by the relation ct=f.*ct-1+i.*c', where .* denotes element wise multiplication. The relation shows the proportion of previous memory and the current memory that is passed on
- Lastly, veiled state St on the time step t is restructured as St =o.*ct. o agrees the amount of the updated recollection component ct which is used as the output of the whole cubicle.

All these weights are updated through time. By Learning these weights for the three data gates, the network models the long term needs.

In order to use LSTM RNN for forecasting of solar irradiance, the data frame is divided into inputs and output so that the inputs can be fed to the deep Learning model to predict the output. Temperature, Relative Humidity, Precipitation and wind speed are taken as input and Solar Irradiation is considered to be the output. The normalised data set is divided into training, validation and testing data set with 624,107 and 304 entries respectively. The input shape is 24 into 5 matrix with batch size of 32. The loss function that has been used is mean square error. The learning scheduler has been used to update the learning rate at every epoch and applies the updated learning rate at the optimizer. Early stopping function has been employed to stop the training in between in order to stop the model get over fitted. The network is developed using Tensor flow and keras Library.

LSTM takes the input shape and batch size and gives an output of dimension 24*350.This is flattened to 8400 *1 dimension. Flatten Layer reshapes the tensor to number of elements in the tensor. Successive dene layer is used for outputting the prediction. The optimizer used for optimisation is Adam optimizer with learning rate of 8e^ (-3). The plot of the loss history and RMSE for LSTM is shown below.

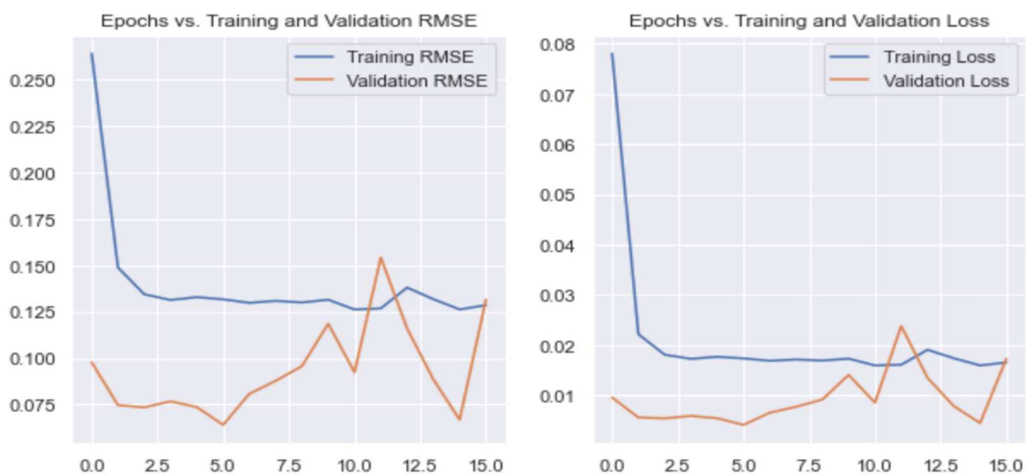


Figure24: Epochs Vs Loss and RMSE for LSTM

The RMSE for LSTM comes out to be 1.285. The predicted solar irradiation and actual solar irradiation for the year 2021 with respect to number of test samples is plotted.

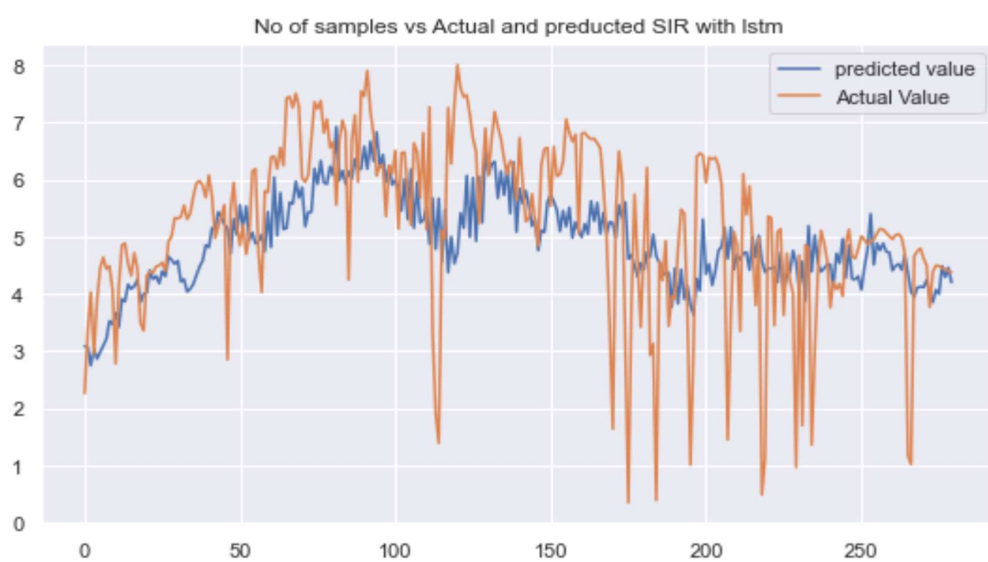


Figure 25: Predicted and Actual Solar Irradiation (LSTM)

5.2 Sequence to Sequence Learning (S2S)

S2S is also known as Encoder Decoder model. The Encoder Decoder model is a way of organising the RNN Sequence to Sequence prediction. As the name suggests it contains an encoder cell and a decoder connected with a context vector. Encoder Decoder Architecture is shown below,

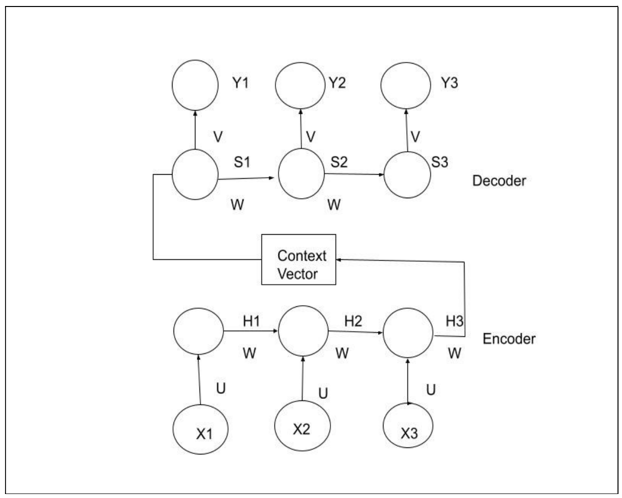


Figure 26: Basic Architecture of Encoder Decoder (S2S)

The encoder is developed by stacking several recurrent units one after another. The recurrent units can be RNN, LSTM or GRU cell. It collects information from the single unit of the input sequence and encodes the information at the end propagates it further. The hidden state is calculated using current input and previous hidden state.

$$h(t)=f(W*h(t-1) + U*x(t))$$

The final hidden state contains all the information from the hidden inputs and the previous current inputs. Context vector links the encoder and the decoder by acting as a final hidden state for the encoder and the initial hidden state for the decoder.

The decoder is also formed by stacking various recurrent units. Each recurrent unit accepts the previous hidden state and produces the current hidden state and the required output.

$$S(t)=f(w*s(t-1))$$

$$Y(t)=v*s(t)$$

Sequence to Sequence model which is s development over LSTM is an encoder decoder model with a hidden context vector. The encoder and decoder contains LSTM cell. The input is first fed to the encoder built of LSTM which outputs a hidden vector, as the focus is on the output of last time step. Repeat Vector layer is used because the output vector is to repeated same number of times as the number of time step in the decoder part. The decoder is constructed with LSTM layer and the parameter return_sequence is kept true, so that each output of the time steps is used by dense layer. Time distributed layer is also used which applies a specific layer such as dense to every sample it takes as input. Successively Flatten and Dense layers are used for outputting the prediction. The optimizer used for optimisation is Adam optimizer with learning rate of 1e^ (-3). The plot of the loss history and RMSE for S2S Algorithm is shown below.

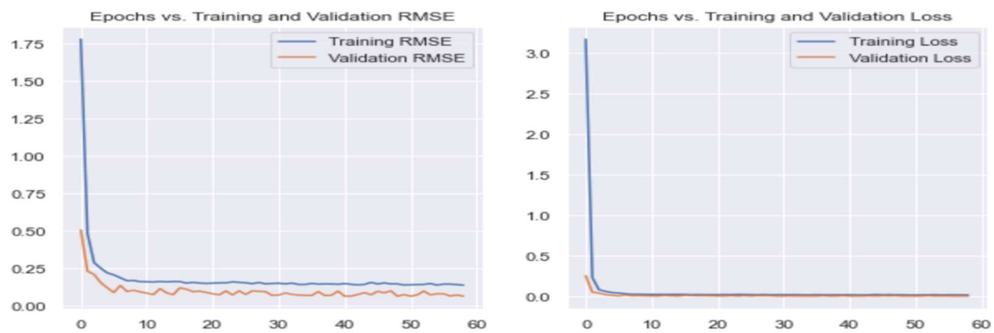


Figure 27: Epochs Vs Loss and RMSE for Encoder and Decoder.

The RMSE for LSTM comes out to be 1.158. The predicted solar irradiation and actual solar irradiation for the year 2021 with respect to number of test samples is plotted.

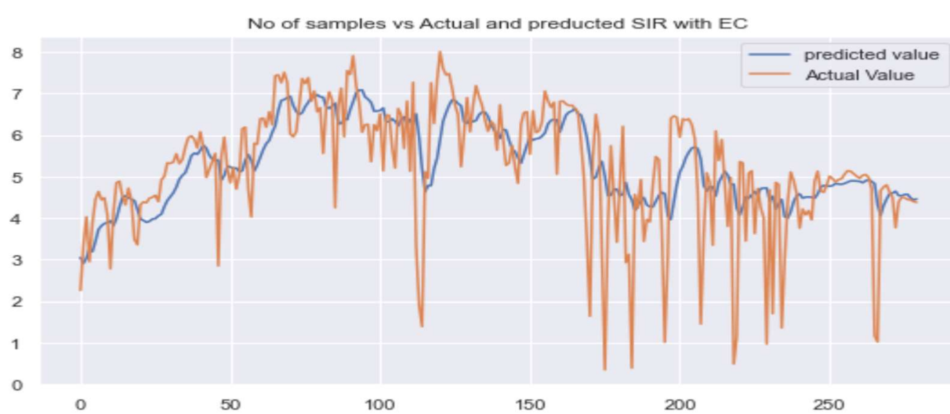


Figure 28: Predicted and Actual Solar Irradiation using Encoder Decoder.

CHAPTER 6

INTERPRETATION

6.1 Comparative Analysis

Machine Learning	Deep Learning
Subset of Artificial Intelligence	Subset of Machine Learning
ML is based on learnings achieved through past activities.	Deep Learning is based on human brain.
These consist of traditional ML and shallow ML algorithms	Consist of interconnected deep layers
Requires less computing power .	Requires more computing power.
Accuracy is more than the traditional programming.	Accuracy more than ML

Table 1: ML Vs Deep Learning.

MATLAB	PYTHON
tool for academic and scientific purposes. It provides the user with an environment for numerical computation. It has its own syntax for writing codes.	It is a general purpose programming language and an interpreter. It consists of standard library for general as well as specific purposes.
Consists of tool kits like signal and data processing, Deep learning and ML tools, tools for engineering and technical purposes. Easy to install and use. Simulink is an exclusive feature of MATLAB.	Consists of packages like numpy, matplotlib, tensor flow, keras. Installation requires features like IDE. Available for free and widely used in data science.

Table 2: MATLAB Vs Python

Algorithm	Platform used	parameters	RMSE	Comment
MLR	MATLAB	regress function is used to calculate the input coefficient matrix.	0.0155	Gives priority to input which has high linearity with output.
SVR	MATLAB	fitrvsm box constraint, epsilon value, polynomial order.	0.0848	Generally used for classification purposes.
FFNN	MATLAB	feedforwardnet hidden layer training function BR algorithm	0.0017	Improvement in accuracy.
LRNN	MATLAB	layrecnet hidden layer training function BR algorithm	6.54 e^{-4}	Accuracy is further improved when past input is also considered.
LSTM	Python	Adam Optimizer	1.285	Development over RNN
S2S	Python	Adam optimizer	1.158	Development over LSTM

Table 3: Comparison of algorithms used in Thesis.

6.2 Conclusion

In recent times most of the distribution companies in India are interested in finding solution to load/demand forecasting. Generally, they collect previous days' hourly consumption data which is provided to Load dispatch centre for next days' demand. Hence they are looking for accurate solution to forecasting through AI and ML. Many DISCOMS are already using AI/ML tool in association with private firm. However, there is mismatch between load forecasting data form AI/ML tool and actual data during weather condition variations and hence there is a need for improvisation. Some of DISCOMS has also established a dedicated smart meter operation centre for smart meter output data to implement in the areas of billing, tampering and revenue area. Further they are interested in accurate load data prediction and other value added services using Smart meter data.

Hence the ML Methodologies used in the thesis for solar irradiation forecasting shall prove to be a basis for demand forecasting of solar power. In this work it has been observed that MLR gives an appropriate result for all KPIs. It is one of the traditional ML algorithm which is widely used for regression. However, it may be impacted by the variables with high correlation and may not give appropriate result with non linearities. SVR which has been in use widely for classification problem does not give satisfactory result when applied for regression. In order to develop FFNN and Layered RNN, number of network architecture has been tried upon to get the optimum result. The levenberg-Marquardt algorithm requires less time but more memory. This is fast while training the neural network. However, when big data sets and large neural networks are present, it requires large memory. Scaled conjugate algorithm requires less memory and is recommended as it uses gradient calculations which are memory efficient. Bayesian Regularisation is a mathematical process that converts a nonlinear regression into a well posed statistical problem in the form of ridge regression.BR algorithm requires more time for training but results in good realisation. All the three algorithms give a satisfactory result, however BR algorithm for FFNN and LM algorithm for RNN has been considered. Among all the algorithm used, RNN with LM algorithm gives the best result on MATLAB. FFNN and MLR also gives accurate prediction.

Further the thesis uses deep learning algorithms like LSTM and S2S for solar irradiance forecasting. The deep learning model has been developed on python. Python is an open source tool and python community provides immense support to users. Python has proved to be immensely beneficial to work on data at the initial stage. Syntax Used in Python is easy to understand and code. It Consists of Libraries which are used for specific task. The Integrated Development Environment used here is jupyter. Libraries that have been used are Tensor flow and Keras for the development of Deep Learning Model.

This thesis has shown that the limitations of RNN can be overcome by using LSTM and further it has proposed the usage of Encoder Decoder Model for forecasting as it is an improvement over LSTM and other recurrent neural networks. It addresses the issue of different length of input sequences and the output sequences. Encoder Decoder can map sequences of different length to each other. The improved RMSE for Encoder Decoder testifies that it can used for time series forecasting. Though RNNs are widely used for Natural Language Processing, the thesis exemplifies its usage in time series forecasting.

6.3 Future Scope of Study

With increasing share of renewable energy in power generation, it is being increasingly important to have accurate solution to load forecasting because of intermittent nature of solar energy in specific. Hence it is very much evident that we shall look towards AI, ML, Deep Learning for accuracy and more reliable solution. Traditional AI and shallow ML algorithms have been in use since long time and hence moving towards deep learning algorithm shall prove to be a boon to power industry. It shall help in getting solution to the issues like demand/load forecasting, AT&C losses, Energy Theft Detection, Prediction of Distribution Transformer failure, Asset Inspection, Vegetation Management, Consumer Experience Enhancement, Renewable Energy integration, power purchase cost optimization, Electricity Market and etc.

Further Transformer network, which is another type sequential algorithm can be brought into use for finding solutions to problems in power systems. Transformers might not be our first choice for time series forecasting but due its versatility and wide range of application, it needs much attention. Further Bayesian neural network can be brought in application as it combines both deep learning and Bayesian. It makes available a deep learning basis that can accomplish high level performance and simultaneously understand and figure out doubt. One of the biggest challenge in deep learning is training data. Deep learning needs huge data set to outperform traditional model and hence it cannot learn fast from a handful of data samples. Hence it suffers from sample efficiency and transferability. Hence meta learning can be helpful when we don’t have enough training data. It tries to copycat learning process of human being by using a past lesson that has been learned from a dissemination of tasks. It gives high performance even with minimum quantity of data.

REFERENCES

[1] Cecilia Martínez-Castillo1, Gonzalo Astray1,2, Juan Carlos Mejuto2, "*Modelling and Prediction of monthly global Irradiation using Machine Learning models*". https://doi.org/10.20944/preprints202103.0573.v1

[2] DEBIRUPA HORE, "*Application of Artificial Intelligence and Machine Learning in The Field of Electrical Engineering: An Overview",* International Journal of Advances in Science Engineering and Technology, ISSN(p): 2321 – 8991, ISSN(e): 2321 –9009 Volume-6, Issue-4, Oct.-2018, http://iraj.in.

[3] E. Haque et al.: "*Comparative Analysis of Deep Neural Networks for Hourly Temperature Forecasting",* IEEE Access, Digital Object Identifier 10.1109/ACCESS.2021.3131533

[4] Huaizhi Wang, Zhenxing Lei, Xian Zhang, Bin Zhou, Jianchun Peng, "*A review of deep learning for renewable energy forecasting"*, Energy Conversion and Management 198 (2019) 111799

[5] Jayati Vaish,Stuti Shukla Datta,Seethalekshmi K, "*Short Term Load Forecasting using ANN and Ensemble Models Considering Solar Irradiance*",International Conference on Electrical and Electronics Engineering (ICE3-2020) 978-1-7281-5846-4/20/$31.00 ©2020 IEEE

[6] Maitha Al Shamisi,Hassan Hejase,Ali H.Assi, "*Using MATLAB to Develop Artificial Neural Network Models Predicting Global Solar Radiation in AI Ain City UAE*", https://doi.org/10.5772/25213/j.rene.221917897.

[7] M.M.Tripathi,K.G Upadhya and S.N Singh, "*Short Term Forecasting Using Generalised Regression and Probabilistic Neural Network in the Electricity Market*", November 2008, Volume 21, Issue 9, The Electricity Journal

[8] Mandeep Singh, Raman Maini, "Various Electricity Load Forecasting Techniques with Pros and Cons", International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-6, March 2020

[9] M. Abuella and B. Chowdhury, "Solar Power Forecasting Using Support Vector Regression," in Proceedings of the American Society for Engineering Management International Annual Conference, 2016.

[10] Mahmoud A. HAMMAD[1], Borut JEREB[2], Bojan Rosi[2], Dejan DRAGAN[2], *Methods and Models for Electric Load Forecasting: A Comprehensive Review*, Logistics & Sustainable Transport Vol. 11, No. 1, February 2020, 51-76 doi: 10.2478/jlst-2020-0004.

[11] Min Xia, Member, IEEE, Haidong Shao, Xiandong Ma, and Clarence W. de Silva, Fellow, IEEE, A Stacked GRU-RNN-Based Approach for Predicting Renewable Energy and Electricity Load for Smart Grid Operation, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 17, NO. 10, OCTOBER 2021

[12] Muhammad Iqbal, Feras Al-Obeidat, Fahad Maqbool, Saad Razzaq, Sajid Anwar, Abdallah Tubaishat, Muhammad Shahrose Khan, and Babar Shah, *COVID-19 Patient Count Prediction Using LSTM*, IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, VOL. 8, NO. 4, AUGUST 2021

[13] Mehmet Sahin, "*Comparison of ANN and MLR models for estimating solar radiation in Turkey using NOAA/AVHRR data",* Advances in Space Research 51(2003) 891-904.

[14] Neural Network Toolbox User's Guide, The Math works, Howard Demuth and Mark Beale.

[15] RUNHAI JIAO, TIANMING ZHANG, YIZHI JIANG, AND HUI HE, Short-Term Non-Residential Load Forecasting Based on Multiple Sequences LSTM Recurrent Neural Network, IEEE Access, Digital Object Identifier 10.1109/ACCESS.2018.2873712.

[16] Samsher Kadir Sheikh,M. G. Unde, "*Short-Term Load Forecasting Using ANN Technique* " , ISSN: 2231 – 6604 doi: 10.7323/ijeset/v1_i2_12.

[17] Swati Sharda, Member, IEEE, Mukhtiar Singh, Member, IEEE, and Kapil Sharma, Senior Member, IEEE, "*RSAM: Robust Self-Attention Based Multi-Horizon Model for Solar Irradiance Forecasting",* IEEE TRANSACTIONS ON SUSTAINABLE ENERGY, VOL. 12, NO. 2, APRIL 2021

[18] Tendani Mutavhatsindi, Caston Sigauke and Rendani Mbuvha, *Forecasting Hourly Global Horizontal Solar Irradiance in South Africa Using Machine Learning Models*, IEEE Access, Digital Object Identifier 10.1109/ACCESS.2020.3034690

[19] Thanasis G. Barbounis, John B. Theocharis, Member, IEEE, Minas C. Alexiadis, and Petros S. Dokopoulos, Member, IEEE, "*Long-Term Wind Speed and Power Forecasting Using Local Recurrent Neural Network Models"*, IEEE TRANSACTIONS ON ENERGY CONVERSION, VOL. 21, NO. 1, MARCH 2006