Islamic University of Technology, Bangladesh

Solar Irradiance Forecasting: Deep Learning Approach

By

Samiur Rashid Abir (180021213)
Mahin Khan Mahadi (180021221)
Al-Muzadded Moon (180021233)

A Dissertation
Submitted in consideration of Partial Fulfillment of the Requirement for the
Bachelor of Science in Electrical and Electronic Engineering
Academic Year: 2021-2022

Department of Electrical and Electronic Engineering (EEE)
Islamic University of Technology (IUT)
The Organization of Islamic Cooperation (OIC)
Gazipur-1704, Dhaka, Bangladesh

Solar Irradiance Forecasting: Deep Learning Approach

A thesis presented to
The Academic Faculty
by

Samiur Rashid Abir (180021213)
Mahin Khan Mahadi (180021221)
Al-Muzadded Moon (180021233)

Approved by

Mr. Fahim Faisal

Ffaisal 26/05/23

....

Fahim Faisal

Assistant Professor

Department of Electrical & Electronic Engineering

Islamic University of Technology (IUT)

The Organization of Islamic Cooperation (OIC)

Gazipur-1704, Dhaka, Bangladesh

Declaration

This is to certify that the thesis entitled **"Solar Irradiance Forecasting: Deep Learning Approach"** is supervised by Mr. Fahim Faisal. This project work has not been submitted anywhere for a degree.

FFaisal 26/05/23

....

Fahim Faisal

Assistant Professor

Dept. Of Electrical and Electronic Engineering

....

Samiur Rashid Abir (180021213)

MahinKhan Mahadi

....

Mahin Khan Mahadi (180021221)

....

Al-Muzadded Moon (180021233)

Acknowledgment

First and foremost, we’d like to thank our respected supervisors for guiding us through this process with the utmost patience and consistency. Even though there were a lot of unknowns and problems, they were always there to help and guide. We learned a lot about this subject from their insightful discussions in different fields.

Then, we’d like to thank our senior brother Sadman Sakib for his help and support, as well as all of the EEE, IUT faculty.

Last but not least, we’d like to say a big thank you to everyone who has helped us, especially our parents and friends who have helped us with their vast knowledge when we’ve had problems. No matter how small the help was, whether it was through inspiration or direct acts of service, we are very thankful.

Abstract

As Bangladesh is situated in a subtropical zone, it is more likely that solar panels installed there will have a high output. This is because of the country's proximity to the equator. This is because there is a greater amount of sunshine during certain times of the year than there is during other times. Forecasting the amount of solar irradiation received is an essential part of photovoltaic systems that are connected to the grid. In order for these systems to make a positive contribution to the equilibrium of supply and demand in the energy market, it is necessary for them to take into account the unpredictability and volatility of the electricity generated by solar panels. As a consequence of this, possessing precise estimations of the irradiation that comes from the sun is something that is something that is extremely vital. Solar irradiation is notoriously difficult to predict and is prone to considerable changes in intensity due to the fact that it is sensitive to a diverse range of atmospheric variables. This sensitivity makes solar irradiation liable to significant swings in intensity.

When forecasting models are forced to generate forecasts regarding sun irradiance several steps into the future, they have a difficult time capturing the long-term sequential correlations that emerge. This is because the sun's irradiance changes with time. As a result, forecasters will have a more difficult time producing reliable forecasts. Because of their ability to deduce long-term relationships from sequential information, attention-based models find widespread use in the study of natural language processing (Natural Language Processing). This study aims to construct, on the basis of an attention model, a structure for the multivariate time series forecasting that can be utilised in practise.

The objective of this study is to develop the structure. The ability of the Attention-based encoder-decoder, Transformer, and Temporal Fusion Transformer (TFT) models to create predictions over a wide range of time horizons is taught,

evaluated, and compared with the capabilities of other forecasting models. The training data, test data, and comparison data for these models are all derived from a total of two separate locations in Bangladesh. It has been demonstrated that TFT is more reliable and accurate than the other algorithms. This was discovered to be the case after it was revealed that attention-based models significantly improved the accuracy of predictions. The findings that we obtained from our research serve as the basis for our conclusions.

Contents

List of Figures

List of Tables

1.Introduction

A significant number of nations all over the globe are now facing a formidable obstacle in the form of the worldwide energy crisis [1]. Within the context of applications in the electrical sector, renewable energy resources have emerged as key participants in the endeavor to strike a balance between energy production and demand [2]. This is an important step toward achieving the goal of achieving a sustainable energy future. Research interests are mostly concentrated on developing new sources of renewable energy as well as enhancing existing ones. Solar power is one kind of renewable energy that may be obtained with relative ease on our planet [3, 4].

The amount of carbon-free power generated by renewable sources, such as the sun, has been steadily rising over the last five years [5, 6]. Many forms of renewable energy have the potential to be beneficial but still need more research and development. Several technologies, such as photovoltaic panels, concentrated solar power plants, and solar water heating systems, may be used to harvest the sun's energy and convert it into a usable form for human use. PV systems are a wonderful alternative for general use because of their modifiability and flexibility, which makes them an excellent choice for widespread deployment in transmission grids, medium-voltage distribution feeders, and low-voltage distribution grids [7]. This makes PV systems an excellent choice for widespread deployment in low-voltage distribution grids.

The array of photovoltaic cells that make up a photovoltaic module is connected so that the total quantity of electricity produced is increased. There is a one-to-one relationship between the amount of solar irradiation that is received and the quantity of electricity that is generated [8–10] Latitude, altitude, weather, cloud cover, humidity, and the basic shifts that occur during the seasons are just some of the elements that may influence the quantity of radiation that is received in various regions of the earth. Solar irradiation may be difficult to forecast, which can make it difficult for those in charge of managing the power grid to keep it under control.

Electric grid management should do a better job of anticipating future energy requirements. If they had access to reliable estimates of future amounts of solar radiation [11], they could schedule their power production to correspond with seasonal shifts in the amount of energy they need. They would thus be able to swiftly react to changes in the amount of electricity that was required as a result of this. To effectively incorporate solar panels into existing electrical grid systems, it is necessary to have information about the irradiance of the sun at a certain location and time in the future [12]. These data are required to make accurate forecasts about the amount of energy that will be produced and delivered.

Using predictions of solar irradiance, an unavoidable process, may allow power system engineers to better monitor peak power production and preserve grid stability [13, 14]. Because of this, solar forecasting on a daily basis has become increasingly fascinating to scientists over the course of the last several decades [15]. This is a direct result of what has been happening. Businesses are spending a significant amount of money on power control systems in order to improve data collecting and autonomous resource management [15, 16]. This trend is occurring as the usage of photovoltaics becomes increasingly widespread. The deep learning algorithms were applied to the national solar radiance dataset in this specific paper and the strategy that produced the greatest results while producing the fewest errors was determined.

2.Literature Review & Motivation

The study of E. Akarslan et al. [17] develops five semi-empirical models to forecast hourly solar radiation. Three regions are intentionally chosen, solar data is observed and gathered hourly, and predicting results are compared to models based on the Angstrom-Prescott (A-P) equation. The proposed technique outperforms A-P equation-based models.

B. Schulz et al. [18] propose post-processing for 30 min-6 h ensemble weather predictions of solar irradiation. In two case studies, the suggested models produce probabilistic forecasts in the form of a censored logistic probability distribution with lead times of up to 5 days. Postprocessing improves ensemble forecasts up to 48 hours in advance, correcting the chronic lack of calibration.

In the research of D. Yang et al. [19] a Complete History Persistence Ensemble (CHPeEn) has been proposed to measure all future probabilistic solar forecasts. CH-PeEn creates empirical distributions of the expected clear-sky index that are time-independent by examining all data. The CH-PeEn continuous ranked probability score (CRPS) is unaffected by prediction horizon or lead time. CH-PeEn's practically unique CRPS improves skill ratings' interpretability.

The work of V. Le Guen et al. [20] uses fisheye photos to anticipate short-term solar irradiation using deep learning. Based on current video prediction with partial differential equations, this architecture extracts spatio-temporal information modeling cloud movements to reliably estimate sun irradiance. Our technique outperforms recent baselines in video prediction and 5-minute irradiance predictions.

A hybrid forecasting method made by M. Caldas et al. [21], describes and evaluates one-minute averaged sun irradiance one to ten minutes in advance. The forecast uses real-time irradiance data and all-sky pictures. Local adaptations of image processing techniques evaluate cloud mean velocity and predict sun disk cloud cover. Using real-time cloud data and the suggested methodology, solar irradiance is estimated. The model is useful for short-term solar resource forecasting under tough, highly variable solar irradiance circumstances.

The study of H. Wang et al. [22] shows that current soft-computing-based solar irradiance forecasting systems are “black boxes” described by incomprehensible functions like the sigmoid. These functions produce hard-to-interpret predictions. Thus, a unique direct explainable neural network with three layers—two linear and one nonlinear—is created to anticipate solar irradiance. The explainable neural network uses the ridge function to analyze solar feature mapping. Direct neural network training uses back-propagation. Data preparation, parameter fine-tuning, and error estimation are pretraining are required. The explainable neural network’s theoretical ability to uncover nonlinear mapping patterns in solar irradiance allows it to clearly explain the forecasting model’s input-output relationship.

M. Marzouq et al. [23] propose optimal ANN-based machine learning forecasting models to predict solar irradiation. An evolutionary framework based on forecasting history and ANN architecture generates numerous models for varied time horizons up to 6 hours in advance. The models were evaluated in 28 Moroccan cities under different climates. These models can accurately predict solar irradiance in places without data using a zoning scenario. Two more scenarios compare the results. All examples showed good generalization ability.

The suggested model by P. Kumari et al. [24] uses meteorological data from 23 California locales, including temperature, precipitation, relative humidity, and cloud cover. The hybrid LSTM-CNN model first extracts temporal features from time-series solar irradiance data and then extracts spatial features from the correlation matrix of various meteorological variables of the target and its neighbor location. A year, four seasons, and three sky conditions are used to evaluate the model's forecast accuracy. The suggested LSTM-CNN model outperforms smart persistence, support vector machines, artificial neural networks, LSTM, CNN, and other hybrid models with a forecast skill score of 37%–45%.

A unique CEEMDAN-CNN-LSTM model for hourly irradiance forecasting is proposed by B. Gao et al. [25]. To extract data characteristics, full ensemble empirical mode decomposition adaptive noise (CEEMDAN) breaks down historical data into constituent series. Second, a deep learning network based on a convolutional neural network (CNN) and long short-term memory network (LSTM) predicts sun irradiance for the next hour. In this study, CNN-LSTM-based solar irradiance forecasting algorithms are also studied. Four real-world datasets of different climatic types are utilized to evaluate the proposed model. According to multiple comparison studies, the CEEMDAN-CNN-LSTM model can accurately estimate solar irradiance.

X. Huang et al. [26] propose a novel multivariate hybrid deep neural model, WPDCNN-LSTM-MLP, for one-hour sun irradiance forecasting. A complex multi-branch hybrid structure with multi-variable inputs underpins the unique WPD-CNN-LSTM-MLP model. The multi-branch hybrid structure uses wavelet packet decomposition (WPD), convolutional neural network (CNN), long short-term memory (LSTM), and multi-layer perceptron networks (MLP) to process hourly solar irradiance and three climate variables: temperature, relative humidity, and precipitation. The new model accurately captures multi-layer inputs, addresses conventional model weaknesses, and improves predictions. Folsom, Clark, and Denver's data verify the model's accuracy. The WPD-CNN-LSTM-MLP deep learning model outperforms traditional individual back propagation neural networks, support vector machines, recurrent neural networks, LSTM, the climatology-persistence reference forecasts method, and the proposed LSTM-MLP model, CNN-LSTM-MLP model, and WPD-CNNLSTM model in hourly irradiance forecasting.

3.Types of Forecasting Technique

- **Autoregression (AR):**

The autoregression (AR) method is a technique that forecasts the values of a time series in the future by employing a linear combination of the values of the time series in the past. This method is also known as the autocorrelation (AC) method. This technique is also referred to as a linear combination model in some circles. The use of AR models has the potential to achieve higher levels of precision if they make use of aspects that are exogenous to the system.

- **Moving Average (MA):**

As a component of this strategy, the error term of a time series is handled as a linear mixing of the several error terms that came before it. This is done in order to reduce the amount of time spent analysing the data. The accuracy of the results should increase with this strategy. Exogenous variables have the potential to be incorporated into MA models as well, which is an additional alternative.

- **Autoregressive Integrated Moving Average (ARIMA):**

In this method, the AR and MA models are merged by first differentiating the time series in order to make it stationary, and then applying the AR and MA models to the stationary time series. This makes the time series suitable for modeling using the combined AR and MA models. The AR and MA models have been combined in this approach to the problem. The final output is a strategy that incorporates aspects of both the AR and MA approaches into its makeup.

- **Seasonal Autoregressive Integrated Moving Average (SARIMA):**

This approach improves upon ARIMA by making it possible for it to take into consideration the seasonal changes that may appear in time series.

- **Exponential Smoothing (ES):**

The time series is presented in the form of a weighted average of prior observations when using the Exponential Smoothing (ES) technique. When employing this methodology, a greater amount of importance is placed on more current data in contrast to that of more historical data. This method is sometimes referred to as the exponential smoothing strategy, particularly in specific circles. There are several different iterations of ES, the most prominent of which are the Holt-Winters Exponential Smoothing (HWES), the State Space Exponential Smoothing (SSES), and the Simple Exponential Smoothing (SES) iterations (SSM).

- **Prophet:**

The generalised additive model (GAM) that forms the foundation of Facebook's Prophet method of prediction is derived from a generalised additive model (GAM), which also acts as the methodology's basis. Since it can take into account seasonality, trends, and the impacts of holidays, it is highly beneficial for time series that contain a number of different seasonalities. This is because it can take into account seasonality. As a consequence of this, it is particularly helpful for time series that contain a great deal of different seasonalities.

- **Deep Learning Models:**

Neural network models with names like "Recurrent Neural Networks" (RNN), "Long Short-Term Memory" (LSTM), and "Gated Recurrent Unit" (GRU) all have the potential to be helpful when it comes to the goal of time series forecasting. These models are set up to deal with the intricate patterns and correlations that may be found in time series data. They can do this because they are equipped to handle such complexity.

4.Methodology

4.1 Dataset Description

The National Solar Radiation Database (NSRDB) was contacted for the purpose of this inquiry in order to get historical data on irradiance for the years 2019 and 2020, from January through December. This information was used to help guide this investigation. The process of modeling and testing the system made use of these data in several ways. In order to evaluate the effectiveness of the models, it is required to look at data that comes from a range of different sources. Both Dhaka and Cox's Bazar in Bangladesh were investigated for the purpose of this research. Dhaka's coordinates are 23.84 degrees north and 90.41 degrees east, and Cox's Bazar's are 21.46 degrees north and 92.01 degrees east. Table 4.1 illustrates the statistical features that are associated with the data that was gathered from these two sites.

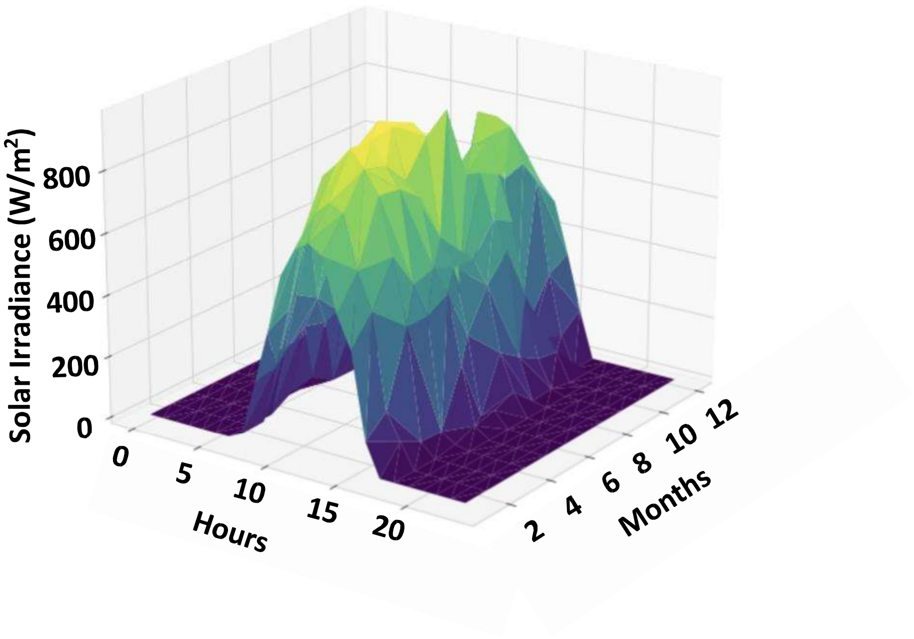
The dataset contains a total of 70176 data points, has a temporal accuracy of 30 minutes, and stores the data in two separate places. There are no blanks in the data that need to be stuffed with something else. The Global Horizontal Irradiation sometimes referred to as GHI, will serve as the principal point of concentration for our experiment. This database includes all three aspects of solar irradiation; one of those aspects is the Global Heating Index (GHI). Figure 4.1 depicts the Global Horizontal Irradiation distribution for the city of Dhaka across the various months of 2019. The graph indicates that the amount of sunshine that is received at its highest level occurs at various times during the day and at various periods throughout the month.

The amount of solar irradiation that is received at any given site is highly dependent on a wide variety of meteorological factors, and these factors can vary substantially from place to place. The amount of solar irradiation that is received from the sun is notoriously difficult to anticipate and undergoes significant shifts if there is cloud cover or precipitation in the sky.

Table i: Statistical features of the solar irradiance data

Location	GHI(W/m^2)		
	Max	**Mean**	**Std.**
All samples	1017	207.23	287.50
Dhaka	994	200.24	278.47
Cox's Bazar	1017	214.23	296.09

The quantity of solar irradiance that is received throughout the day is shown in Figure 4.2 for two distinct kinds of weather: a clear sky and an overcast sky. When there is no cloud in the sky, a pattern may be seen in the data. When clouds are present, however, the GHI measurements become exceedingly difficult to anticipate and display a precipitous decline in the curve.

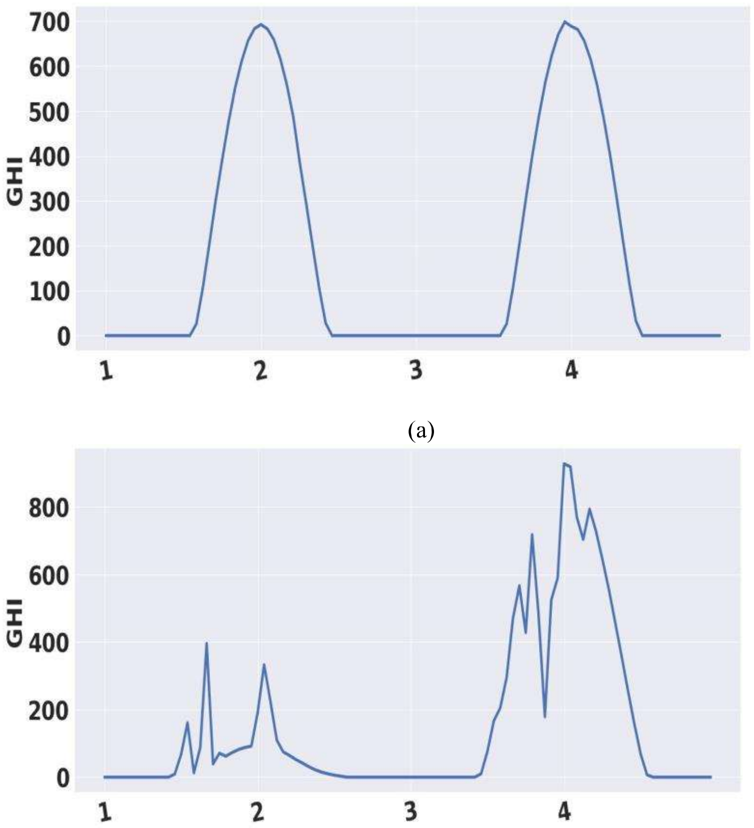


i. Figure 4.1: Solar irradiation data in Dhaka during 2019

In order to improve the accuracy of our model's forecasts, in addition to the data on solar irradiance, the meteorological data that is included in the National Solar Radiation Database is also employed. The meteorological data and their generation are broken out in Table 4.2, which can be seen here.

Table ii: Meteorological parameters

Variable name	Unit
Global Horizontal Irradiance	W/m^2
Ozone	
Solar Zenith Angle	Degree
Precipitable Water	cm
Temperature	℃
Dew Point	℃
Relative Humidity	%
Pressure	mbar
Wind Direction	Degrees
Wind Speed	m/s



(a)

ii. Figure 4.2: Global Horizontal Irradiation during (a) clear-sky (b) cloudy day

4.2 Data Preprocessing

4.2.1 Feature Selection

Weather conditions of many different varieties have the potential to modify solar radiation as it travels through the atmosphere. It is essential to sort the weather-related qualities into those that are useful to the model and those that are irrelevant to the model in order to find the most suitable subset of characteristics to utilise as the model's input. This will make it possible to select the most appropriate feature to serve as the input to the model. The Pearson correlation coefficient is a way of assessing, from a statistical point of view, the degree to which two continuous variables are related to one another. This may be done by comparing the correlation between the two variables. A check was done on the link that existed between the GHI and other meteorological variables in order to identify which components need to be used as inputs. This check was done in order to determine which components need to be utilised as inputs. Table 4.3, which can be seen here, displays the Pearson correlation coefficients that were calculated for the solar irradiance and meteorological variables that were included in the dataset.

The fact that the connection between GHI and the various weather conditions differs depending on the location is one of the reasons that determine the values of these parameters. The climate also has a part in determining these values. The table demonstrates that Temperature, Humidity, Solar Zenith Angle, and Wind Speed were believed to be crucial for the model, however, the other parameters were removed since they did not have a strong association with the GHI. The reason for this was the fact that the other factors did not have a strong relationship with the GHI. When compared to the values of Temperature, Humidity, Solar Zenith Angle, and Wind Speed, this can be observed to be the case.

4.2.2 Feature Transform and Encoding

A classification system that puts together different cloud circumstances and forms of weather is referred to as a "cloud type," and the phrase "cloud type" refers to this classification system. The understanding of this phenomenon is vital since clouds are responsible for the sudden change in the amount of radiation that is received at the surface. One-hot encoding is used to express this attribute as there is no preset sequence for it to follow.

Table iii: Pearson's correlation coefficients between meteorological parameters and GHI

Weather Variables	Dhaka	Cox's Bazar
Ozone	0.064	0.047
Solar Zenith Angle	-0.815	-0.817
Precipitable Water	-0.002	-0.048
Temperature	0.510	0.271
Dew Point	0.018	-0.021
Relative Humidity	-0.547	-0.470
Surface Albedo	0.083	-0.015
Pressure	-0.007	0.057
Wind Direction	0.054	0.093
Wind Speed	0.227	-0.033

Figure 4.1 provides a crystal-clear illustration of the association that exists between GHI and the passage of time. The Date Time variable is an additional component that is of the utmost importance. Due to the presence of an excessive number of categories, one-hot encoding will not operate correctly when combined with this capability. In addition, the connections between the variables are set up in

such a way that prohibits one-hot encoding from being relevant to the circumstances. Even while December and January appear to be 11 months apart from one another, they are actually only separated by one month from one another. In order to find a solution to this problem, we are going to encode the cyclic characteristic by applying the sine and cosine transforms.

4.2.3 Data Scaling and Splitting

If the scales of the continuous input variables are not consistent with one another, the learning process of the system may be significantly slowed down or it may become stuck in the presence of local optimums.

In the case that the size or distribution of the time series data does not change, algorithms like neural networks that employ gradient descent will perform more efficiently. This is because gradient descent is used to find the optimal solution to a problem. This demonstrates that the data need to be normalised in order to make certain that every attribute has the same size and weight. In order to make the results of this inquiry comparable to one another, the data were subjected to a process known as standardization (z-score), which involves rescaling the distribution of values such that they have a mean of 0 and a standard deviation of 1.

In other words, the mean of the values was set to 0, and the standard deviation was set to 1. The following is the formula for the normalisation of z-scores:

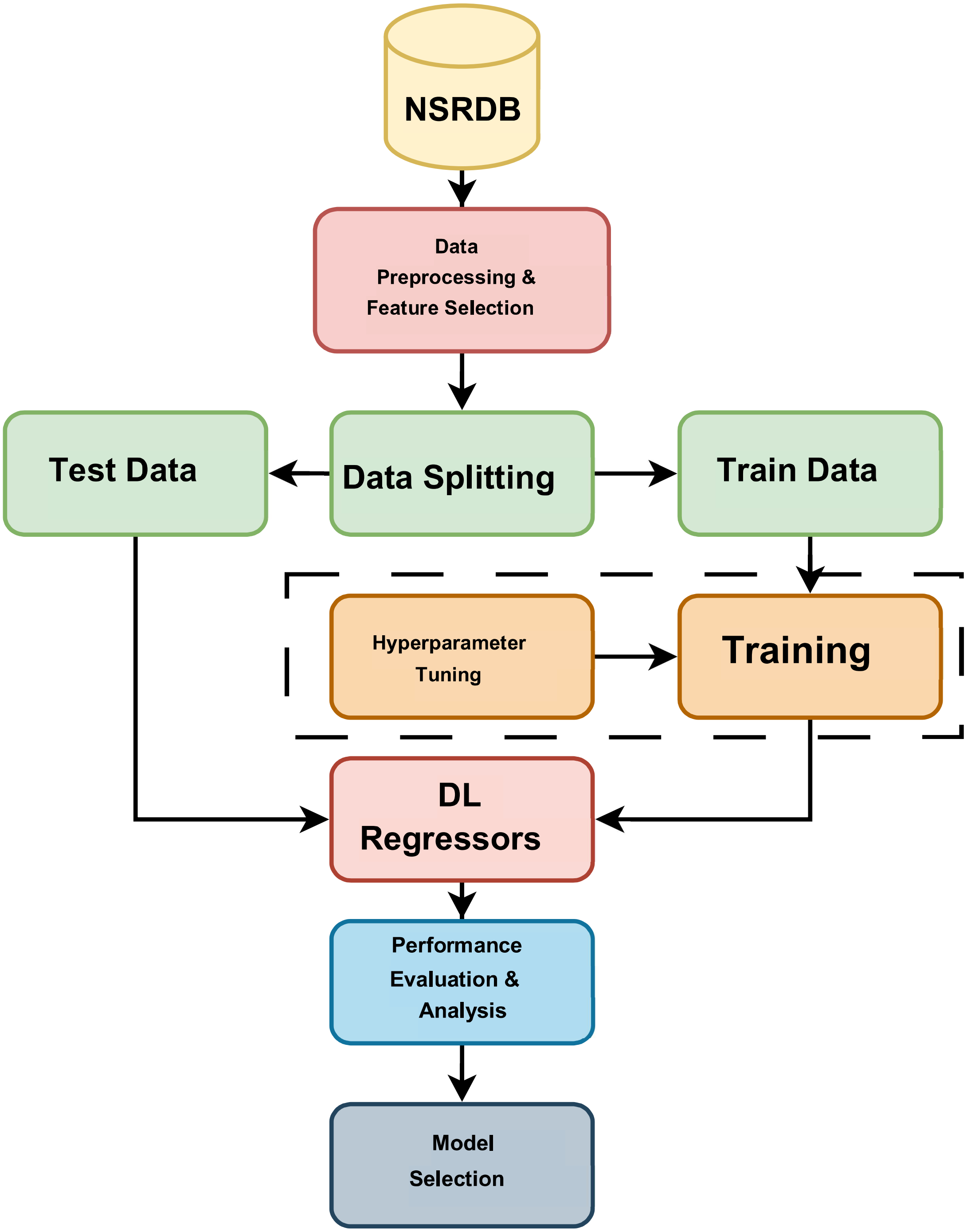
$$x_z = \frac{x_i - \bar{x}}{\delta} \tag{4.1}$$

Where x_i is the input data, ¯x denotes the mean of the feature vector and σ denotes the feature vector’s standard deviation.

For the purposes of training, the entire dataset is divided up into three different groups: the train set, the validation set, and the test set. The models are trained to the data included in the training set, which accounts for 75 percent of the total data. It covers the first year, which is 2019, as well as the first half of the next year, which is 2020. The most recent six months include a validation set and a test set that each account for 12.5% of the total. These most recent six months are the most current. A bias-free evaluation of the model's performance is carried out with the help of the validation set throughout the process of fine-tuning the hyperparameters of the fitted model. The findings of the test set are used as the basis for the evaluation of the final model.

When time series data are divided, the data points do not become jumbled because the order in which they were gathered must be retained. This ensures that the data points do not become confused.

4.3 Overall Workflow



iii. Figure 4.3: Overall workflow

5.Study of Algorithms:

5.1 Long Short-Term Memory (LSTM):

The Long Short-Term Memory (LSTM) is a recurrent neural network (RNN) architecture that has been specifically developed to tackle the vanishing gradient problem, a significant obstacle encountered by conventional RNNs. The phenomenon known as the vanishing gradient problem arises when gradients that are propagated through time experience an exponential decrease in magnitude, thereby impeding the network's ability to effectively learn long-term dependencies.

LSTM introduces a memory cell that allows the network to selectively remember or forget information over time. The fundamental concept underlying LSTM involves the utilisation of gates that regulate the information flow within the cell. These gates include the input gate, forget gate, and output gate, each of which is responsible for different operations.

The input gate determines which information from the current input should be stored in the memory cell. The forget gate is responsible for determining the information to be discarded from the previous state. The output gate is responsible for controlling the output by considering the present input and the state of the memory cell.

The utilisation of these gates enables the Long Short-Term Memory (LSTM) model to effectively retain significant information over extended periods while filtering out irrelevant data. This ability makes it well-suited for tasks that require capturing and understanding long-term dependencies, such as speech recognition, natural language processing, and time series analysis.

In the realm of deep learning, LSTM has demonstrated significant efficacy as a tool, enabling neural networks to proficiently model and acquire knowledge from sequential data. Its ability to address the vanishing gradient problem and handle long-term dependencies has made it a popular choice for various applications in both research and industry.

The Long Short-Term Memory (LSTM) model comprises of recurrent units referred to as cells that retain a hidden state and a memory cell state. The memory cell functions as a storage device, retaining data for extended durations. The primary innovation of Long Short-Term Memory (LSTM) is the incorporation of gating mechanisms that regulate the inflow and outflow of information to and from the memory cell.

The input gate, denoted as "i," is responsible for determining the specific segments of the current input that are to be retained in the memory cell. It calculates a vector that represents the relevance of each input element. The input gate takes into account the current input, as well as the previous hidden state, and passes the relevant information through a sigmoid activation function.

The forget gate, represented as "f," decides what information from the previous memory cell state should be forgotten or discarded. It calculates a vector that determines the importance of each element in the memory cell state. The forget gate takes into account the previous hidden state and the current input and applies a sigmoid activation function.

The state of the memory cell, denoted as "C," is updated through the amalgamation of information from both the input gate and the forget gate. The forget gate is responsible for determining the information that needs to be eliminated from the previous memory cell state, while the input gate is responsible for determining the information that needs to be incorporated. These operations are performed element-wise.

The output gate, denoted as "o," regulates the transmission of data from the memory cell to the present hidden state or the output of the LSTM. The output gate takes into account the current input and the previous hidden state and applies a sigmoid activation function. The updated memory cell state is then passed through a hyperbolic tangent activation function to produce the current hidden state or output.

The utilisation of gating mechanisms enables the Long Short-Term Memory (LSTM) model to selectively retain or discard information during each time step. This capability enables the network to capture long-term dependencies in sequences by preserving relevant information and discarding irrelevant or noisy information.

LSTM networks can be stacked to create deeper architectures, with the output of one LSTM layer serving as the input to the next. Deep Long Short-Term Memory (LSTM) networks have demonstrated enhanced efficacy in diverse applications, owing to their ability to acquire hierarchical representations of sequential data.

To summarise, LSTM is a potent form of recurrent neural networks that tackles the issue of vanishing gradient and facilitates the representation of extended dependencies. The utilisation of gating mechanisms enables LSTM to selectively store, forget, and output information, rendering it highly suitable for tasks that involve sequential data analysis.

One of the key advantages of LSTM is its ability to handle the vanishing gradient problem. In conventional Recurrent Neural Networks (RNNs), the gradients have a tendency to diminish as they propagate in a backward direction through time. This can pose a challenge for the network to effectively learn from lengthy sequences. The Long Short-Term Memory (LSTM) approach resolves this concern by incorporating the memory cell state and gating mechanisms.

The memory cell state functions as a unit for long-term memory in the LSTM model. This facilitates the network to preserve information for prolonged durations, thereby offering a resolution to the issue of the vanishing gradient. The state of the memory cell is updated by means of additive operations that are governed by the input and forget gates. These gates regulate the inflow and outflow of information to and from the memory cell.

The function of the input gate is to regulate the transmission of information from the present input to the memory cell. The process involves determining the specific components of the input that are to be retained in the memory cell state. The forget gate, on the other hand, regulates the retention or removal of information from the previous memory cell state based on the current input. The sigmoid activations of these gates facilitate the ability of LSTM to acquire the skill of retaining pertinent information while discarding irrelevant or redundant information.

The output gate is responsible for regulating the transmission of data from the memory cell to the present hidden state or output. The system adjusts the output in accordance with the present input and the state of the memory cell. The output gate of the Long Short-Term Memory (LSTM) model enables the selective disclosure of information from the memory cell. This ensures that the network prioritises pertinent and valuable output.

LSTM networks have been successful in a wide range of tasks. Long Short-Term Memory (LSTM) models have been applied in various natural language processing tasks, including but not limited to language modelling, machine translation, sentiment analysis, and named entity recognition. Long Short-Term Memory (LSTM) models have been utilised for both acoustic modelling and language modelling in the field of speech recognition. LSTMs have also been applied to time series analysis, including forecasting, anomaly detection, and signal processing.

Furthermore, it is possible to expand LSTM architectures by incorporating multiple layers, thereby producing deep LSTM networks. Deep LSTMs have shown improved performance in capturing complex dependencies and learning hierarchical representations. These architectures facilitate the network in extracting high-level features and abstract representations from sequential data.

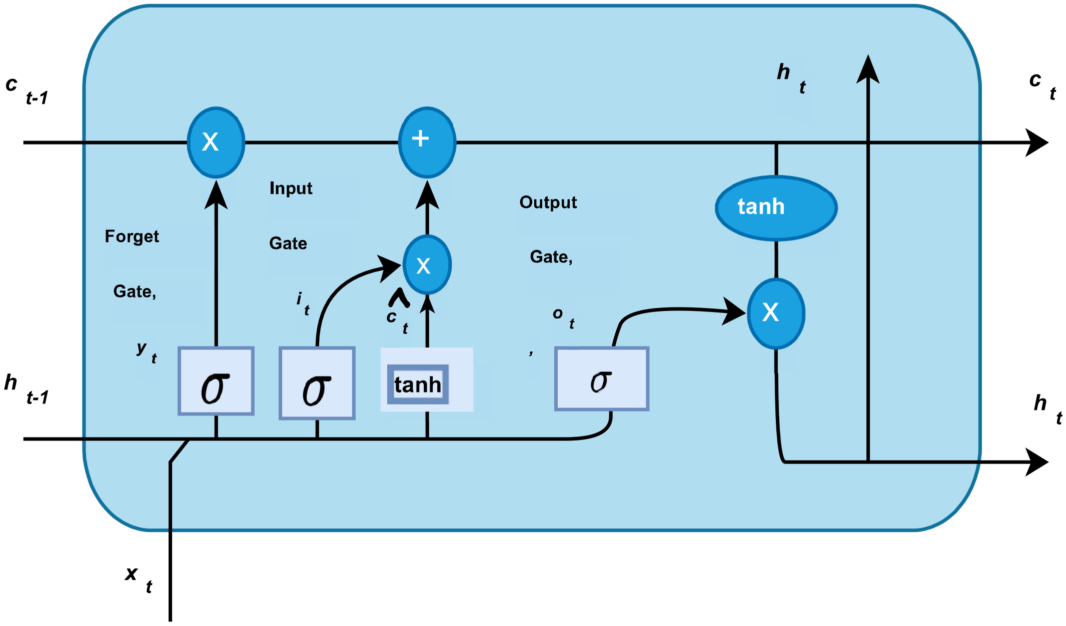
To summarise, LSTM is a potent and commonly utilised form of recurrent neural networks that tackles the issue of vanishing gradient and facilitates the representation of extended-term dependencies. LSTM networks are capable of selectively storing, forgetting, and outputting information by utilising memory cells and gating mechanisms. As a result, they are highly effective for a wide range of tasks that involve the analysis of sequential data.

Long Short-Term Memory (LSTM) networks have garnered significant attention owing to their capacity to manage long-term dependencies, a critical aspect in numerous practical applications. They have shown impressive performance in tasks such as speech recognition, machine translation, sentiment analysis, handwriting recognition, and more.

An essential feature of Long Short-Term Memory (LSTM) is its capacity to dynamically acquire the significance of distinct time steps within a sequence. This adaptability is achieved through the gating mechanisms. The sigmoid and hyperbolic tangent activation functions used in LSTM allow for non-linear transformations that capture complex patterns and relationships in the data.

The input gate and forget gate together determine the information flow into and out of the memory cell. The sigmoid activation function squashes the gate values between 0 and 1, representing the level of importance or relevance. This selective gating mechanism enables LSTM to retain useful information for long periods and discard unnecessary details, making it effective in modeling sequences with long-range dependencies.

The process of training LSTM networks generally entails utilising backpropagation through time (BPTT), which is a variation of backpropagation that expands the recurrent structure for a predetermined number of time steps. The network parameters are updated by computing the gradients. This training process allows the LSTM to learn the optimal values for the weights and biases, enabling it to make accurate predictions or classifications.



iv. Figure 5.1: LSTM structure

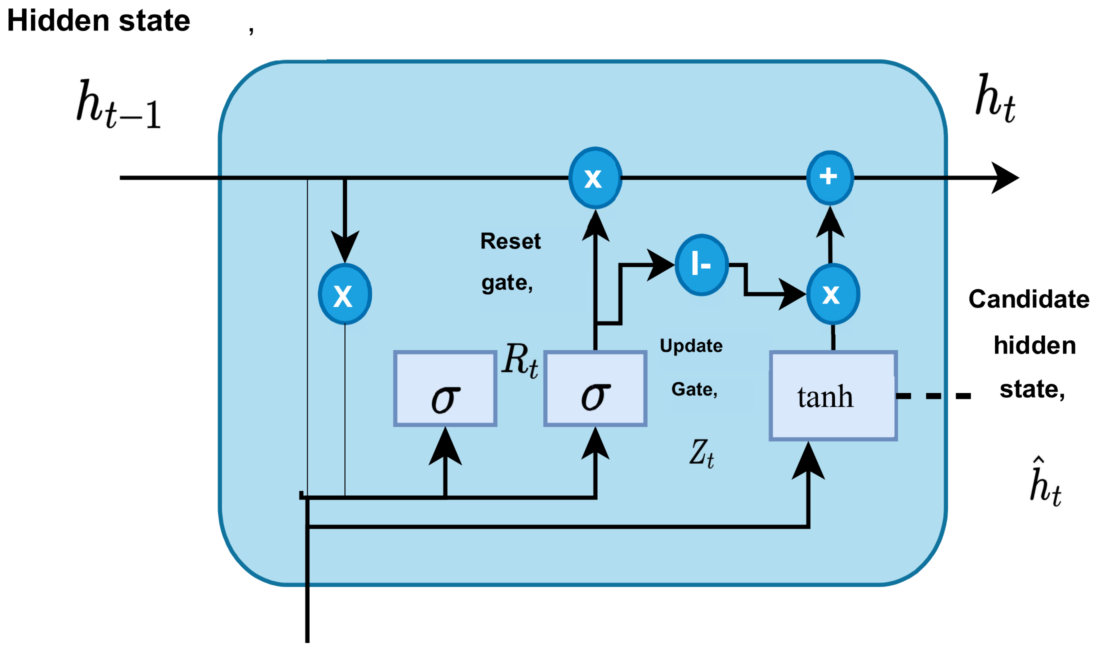
One limitation of LSTMs is their computational complexity, especially in deep architectures. As the number of LSTM layers and units increases, the network's computational requirements grow significantly. To address this, researchers have explored various techniques, such as LSTM variants with reduced computational complexity or using techniques like attention mechanisms to focus on relevant parts of the input sequence.

In recent years, advancements in LSTM have been made to improve its performance and address specific challenges. Variants such as Gated Recurrent Unit (GRU) and Depth Gated LSTM have been introduced, offering alternative architectures with simplified gating mechanisms or improved memory cell structures.

In conclusion, LSTM is a powerful and widely used type of recurrent neural network architecture that addresses the challenges of learning long-term dependencies. Its gating mechanisms and memory cell state allow for adaptive information processing, making it effective in a variety of sequential data tasks. LSTM and its variants remain at the forefront of deep learning and have significantly transformed the field of sequence modelling through continuous research and development.

5.2 Gated Recurrent Unit (GRU):

Gated Recurrent Units (GRUs) represent a recurrent neural network (RNN) architecture that has gained significant traction in the domain of deep learning. Gated Recurrent Units (GRUs) were developed as a means of augmenting conventional Recurrent Neural Networks (RNNs) in order to mitigate the issue of vanishing gradients and enhance the network's learning capacity. This article aims to examine the fundamental principles of GRUs, including their architecture, training methodology, and practical applications.



v. Figure 5.2: GRU structure

Recurrent Neural Networks (RNNs) are a category of neural networks that are specifically engineered to operate with sequential data, such as natural language or time series. In contrast to feedforward neural networks, Recurrent Neural Networks (RNNs) possess connections that establish directed cycles, enabling them to retain hidden states and handle data from preceding time steps. This renders them appropriate for tasks that entail sequential dependencies and comprehension of context.

Traditional Recurrent Neural Networks (RNNs) are known to encounter the vanishing gradient problem, wherein the gradients tend to decrease exponentially over time. Consequently, the ability of Recurrent Neural Networks (RNNs) to learn sequences with long time lags is limited due to the challenge of capturing long-term dependencies. To address this constraint, scholars have introduced various types of RNN variations, such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs).

Gated Recurrent Units (GRUs), similar to Long Short-Term Memory (LSTM) networks, are specifically engineered to tackle the issue of vanishing gradient and facilitate Recurrent Neural Networks (RNNs) in effectively capturing long-term dependencies. This is accomplished by implementing gating mechanisms that discerningly refresh and reset data within the concealed state. The architecture of a Gated Recurrent Unit (GRU) comprises the following essential components:

The update gate regulates the degree to which the preceding hidden state is modified by the present input. It takes the concatenation of the previous hidden state and the current input as input and produces an update gate vector between 0 and 1. A value of zero denotes the absence of an update, whereas a value of one indicates a full update.

The reset gate is a crucial component that governs the degree of omission of the prior hidden state during the computation of the present state. The reset gate vector is produced by inputting the concatenation of the previous hidden state and the current input. A value of zero denotes disregard, whereas a value of one signifies inclusion.

The present memory content denotes the fresh candidate values intended for the concealed state. The computation involves the application of a non-linear activation function, such as the hyperbolic tangent (tanh), to the concatenation of the reset gate and the previous hidden state. This particular step enables the Gated Recurrent Unit (GRU) to selectively update the hidden state based on the current input.

The concealed state of a Gated Recurrent Unit (GRU) signifies the neural network's memory. The current output is determined by a weighted combination of the previous hidden state and the current memory content, as determined by the update gate. The hidden state is responsible for capturing pertinent information from the previous time steps and influencing the output of the GRU at the current time step.

The process of training a GRU entails the optimisation of the model parameters with the aim of minimising a designated loss function. The conventional approach for this task involves utilisation of the backpropagation through time (BPTT) algorithm, which is a variant of the backpropagation algorithm specifically designed for recurrent neural networks.

Throughout the training process, a series of input data is introduced into the Gated Recurrent Unit (GRU), and the neural network generates forecasts at every individual time step. The forecasted outcomes are contrasted with the actual labels, and the difference is measured using an appropriate loss function, such as cross-entropy, for tasks related to classification. The backpropagation algorithm is utilised to calculate the gradients of the loss function in relation to the parameters of the GRU.

In order to modify the model parameters, an optimisation algorithm, such as stochastic gradient descent (SGD) or one of its variants, is employed. Gradients are utilised to modify the weights and biases of the GRU in the direction opposite to the gradient, with the objective of minimising the loss function. The aforementioned procedure is iterated for multiple epochs or iterations until the model reaches a satisfactory solution.

Gated Recurrent Units (GRUs) present various benefits in comparison to conventional Recurrent Neural Network (RNN) architectures, rendering them a prevalent selection in numerous applications.

In comparison to Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRUs) exhibit a less complex architecture featuring a reduced number of gating mechanisms. The streamlined nature of GRUs frequently leads to expedited training and inference durations, rendering them a more effective option, particularly in scenarios involving extensive datasets or applications requiring real-time processing.

It has been demonstrated that GRUs possess a memory capacity comparable to LSTMs, while necessitating a reduced number of parameters. This feature renders them highly advantageous in situations where computational resources are constrained, owing to their ability to strike a balance between memory capacity and efficiency.

Gated Recurrent Units (GRUs) mitigate the issue of vanishing gradient by incorporating update and reset gates, which facilitate the flow of gradient across time with greater efficiency. GRUs are capable of efficiently capturing long-term dependencies in comparison to conventional RNNs by selectively updating and resetting information in the hidden state.

Gated Recurrent Units (GRUs) have demonstrated a notable ability to achieve satisfactory results even when trained on limited amounts of data. They possess the ability to learn effectively from a restricted set of samples, rendering them appropriate for tasks that pose difficulties in acquiring a vast labelled dataset due to cost or other constraints.

Gated Recurrent Units (GRUs) have demonstrated effective modelling of sequential dependencies in language data for natural language processing tasks such as machine translation, text generation, sentiment analysis, and speech recognition. They have demonstrated significant utility in tasks that entail lengthy sentences or documents.

Gated Recurrent Units (GRUs) are proficient in capturing temporal dependencies in time series data, rendering them appropriate for tasks such as stock market prediction, weather forecasting, and anomaly detection. They are capable of effectively learning from historical patterns and making predictions based on the acquired temporal context.

Although GRUs are primarily intended for sequential data, they can also be utilised for image and video analysis tasks. By treating images or video frames as sequential data, GRUs can model the dependencies between adjacent frames and perform tasks like action recognition, video captioning, and video generation.

Gated Recurrent Units (GRUs) have been employed in recommender systems to effectively capture and analyse user behaviour patterns over a period of time. GRUs have the capability to analyse the sequential interactions of users with items, enabling them to capture evolving preferences and provide personalised recommendations based on learned patterns.

Gated Recurrent Units (GRUs) are a robust variation of recurrent neural networks that have been developed to overcome the issue of vanishing gradients. They have been shown to enhance the learning capabilities of conventional RNNs. GRUs have become increasingly popular in a variety of applications, including natural language processing, time series analysis, image and video analysis, and recommender systems, due to their efficient architecture, effective gradient flow, and capacity to capture long-term dependencies.

As the field of deep learning progresses, it is expected that GRUs, along with other variants of RNNs, will continue to play a crucial role in the deep learning toolkit. This will facilitate the creation of more advanced models for processing sequential data, thereby enhancing the capabilities of AI systems.

5.3 Seq2seq Encoder Decoder:

Seq2Seq (Sequence-to-Sequence) models have revolutionized various natural language processing tasks such as machine translation, text summarization, and chatbot development. The encoder-decoder framework is a fundamental architecture utilised in Seq2Seq models. This article will provide an in-depth analysis of the Seq2Seq encoder-decoder architecture, examining its constituent elements, training methodology, and practical uses.

Understanding Seq2Seq Models: a. Seq2Seq Architecture Overview:

The Sequence-to-Sequence (Seq2Seq) model comprises of two primary constituents, namely an encoder and a decoder. The encoder processes the input sequence, such as a sentence, into a fixed-size vector representation called a context vector. Subsequently, the decoder utilises the aforementioned context vector to produce a sequence of output, which is usually in a distinct language or format.

Encoder: The encoder is typically a recurrent neural network (RNN), such as LSTM (Long Short-Term Memory) or GRU (Gated Recurrent Unit). The input sequence is subjected to iterative processing of each element to capture its contextual information. The final hidden state of the encoder serves as the context vector, summarizing the entire input sequence.

The decoder is an additional recurrent neural network that receives the context vector generated by the encoder and produces the output sequence. The model forecasts the subsequent element in the sequence by utilising the context vector and the previously produced elements. The decoder may incorporate an attention mechanism to concentrate on pertinent segments of the input sequence during the output generation process.

Training Seq2Seq Encoder-Decoder Models: a. Data Preparation: The first step in implementing Seq2Seq Encoder-Decoder models is to prepare the data.

Sequential-to-Sequential (Seq2Seq) models necessitate parallel data, wherein the input sequences and their corresponding target sequences are aligned. For machine translation, this would consist of pairs of source sentences and their translations. The data is preprocessed, tokenized, and transformed into numerical representations before training.

Loss Function: The most common loss function used in Seq2Seq models is the cross-entropy loss. It calculates the difference between the predicted sequence and the target sequence, encouraging the model to generate sequences that closely match the ground truth.

Training Process: During training, the input sequence is fed into the encoder, and the decoder generates the output sequence step by step. The loss is calculated at each step, and backpropagation is performed to update the model's parameters. The utilisation of the teacher forcing technique, which involves providing the decoder with the ground truth output as input during the training process, is a common practise aimed at enhancing the stability of the training process.

During the process of inference, the model that has been trained is utilised to produce sequences. The input sequence is processed by the encoder, and the output is generated by the decoder using the acquired context vector. Decoding can be executed through methodologies such as beam search or sampling.

Applications of Seq2Seq Encoder-Decoder Models: a. Machine Translation: Seq2Seq models have achieved significant success in machine translation tasks. Through the utilisation of extensive parallel corpora training, it is feasible for them to acquire the ability to proficiently translate across diverse languages.

Text summarization models have been utilised to produce brief summaries of lengthy documents or articles. Through the process of encoding the source text and subsequently decoding a condensed summary, one can effectively capture the most salient information.

Chatbots and Conversational AI: Seq2Seq models are used to build chatbots that can engage in human-like conversations. By training on dialogue datasets, they learn to generate appropriate responses based on the input query.

Speech Recognition and Synthesis: Seq2Seq models have been used for speech recognition tasks, converting spoken language into written text. They can also be employed for text-to-speech synthesis, generating natural-sounding speech from written text.

Improvements and Variations of Seq2Seq Encoder-Decoder Models:

Attention Mechanism:

One major improvement to the basic Seq2Seq architecture is the introduction of attention mechanisms. Attention allows the decoder to focus on different parts of the input sequence while generating the output. This feature enhances the model's ability to effectively process lengthy sequences and enhances the quality of translations.

The Bidirectional Encoder is a model where the input sequence is processed in both forward and backward directions, unlike the standard Seq2Seq model where only the forward direction is used for encoding. Utilising a bidirectional encoder that processes the sequence in both forward and backward directions enables the model to capture a more comprehensive range of contextual information. This leads to better performance, especially in tasks where the context from both ends of the sequence is important.

Transformer-based Seq2Seq Models: The Transformer model, introduced in the "Attention is All You Need" paper by Vaswani et al., brought significant advancements in Seq2Seq architectures. Instead of using recurrent neural networks, Transformers utilize self-attention mechanisms, allowing for parallel processing of input sequences. Transformers have demonstrated cutting-edge performance in diverse tasks and are extensively employed in machine translation and other sequence generation endeavours.

The utilisation of reinforcement learning techniques can be employed to refine the generated output, as opposed to the maximum likelihood estimation utilised during training in traditional Seq2Seq models. By using reinforcement learning, models can optimize for specific metrics such as BLEU score or ROUGE score, which are commonly used for evaluating the quality of machine translation or summarization.

Variational Autoencoders (VAEs): VAEs combine the Seq2Seq architecture with latent variable modeling. The integration of a probabilistic encoder-decoder framework enables Variational Autoencoders (VAEs) to produce a range of significant and varied results. VAEs have been applied to tasks like text generation and dialogue systems, where capturing the uncertainty and diversity of the output is desirable.

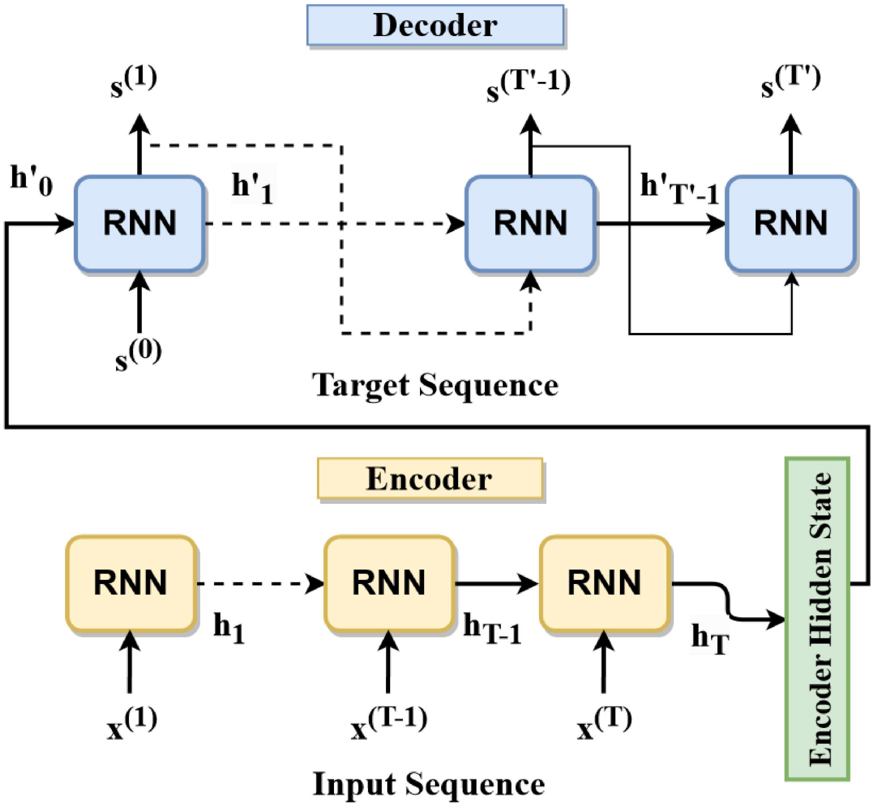
Challenges and limitations:

Management of lengthy sequences: Sequencing-to-sequencing (Seq2Seq) models, particularly those that rely on recurrent neural networks, encounter difficulties when handling extended sequences. The presence of extended dependencies within the input sequence may result in the occurrence of vanishing or exploding gradients, thereby posing a challenge for the model to effectively capture pertinent information. Transformers have partially addressed this issue, but it remains a challenge in certain scenarios.

Words that are not included in a particular language model's vocabulary, also known as Out-of-Vocabulary (OOV) words.

Seq2Seq models struggle with out-of-vocabulary words, which are words not present in the training vocabulary. Handling OOV words requires robust strategies such as using subword tokenization techniques or incorporating external resources like pre-trained word embeddings.

The disambiguation of the correct context can pose a challenge for Seq2Seq models, particularly in tasks such as machine translation or text summarization. The model may generate outputs that are grammatically correct but semantically incorrect due to ambiguous source inputs.



vi. Figure 5.3: Seq2seq Encoder Decoder Structure

Over-Reliance on Input Context: Seq2Seq models heavily rely on the encoded context vector, which means they may fail to generate outputs that are inconsistent with the input context. This constraint has the potential to impede creativity or the ability to adapt to evolving contexts, wherein the model is expected to produce outputs that transcend the limitations of the input.

In conclusion, it can be stated that Seq2Seq encoder-decoder models have significantly transformed the field of natural language processing by facilitating the generation and translation of sequences. Their ability to capture complex dependencies and generate coherent output sequences has made them invaluable in machine translation, text summarization, chatbots, and other applications. Continual research is being conducted to tackle the obstacles and constraints linked with Seq2Seq models, resulting in enhanced structures and methodologies. Seq2Seq encoder-decoder models continue to be at the forefront of natural language processing research and development due to their versatility and wide range of applications.

5.4 Encoder Decoder with Attention Mechanism:

It is possible to produce forecasts about the future values of a time series by using the series' present values in combination with an encoder-decoder that possesses an attention mechanism.

In this example, the encoder network takes the past values of the time series that it gets as input and turns them into a context vector that has a set number of components. The decoder network will construct a list of the time series' future values once it has received the context vector as an input.

Along the same lines as the NLP use case, the attention mechanism is used to provide the decoder with the capacity to pay attention to particular portions of the input sequence at certain time steps. This is accomplished by providing the decoder with the ability to pay attention to certain regions of the input sequence. But in the context of time series forecasting, the attention mechanism may also be used to locate and weigh the relevance of different input elements that may be impacting the time series, such as weather data or stock prices. In other words, the attention mechanism can be used to figure out what factors are affecting the time series and how they are affecting it. In other words, the attention mechanism may be utilised to determine what elements are impacting the time series as well as the manner in which they are affecting it in order to better understand the relationship between the two.

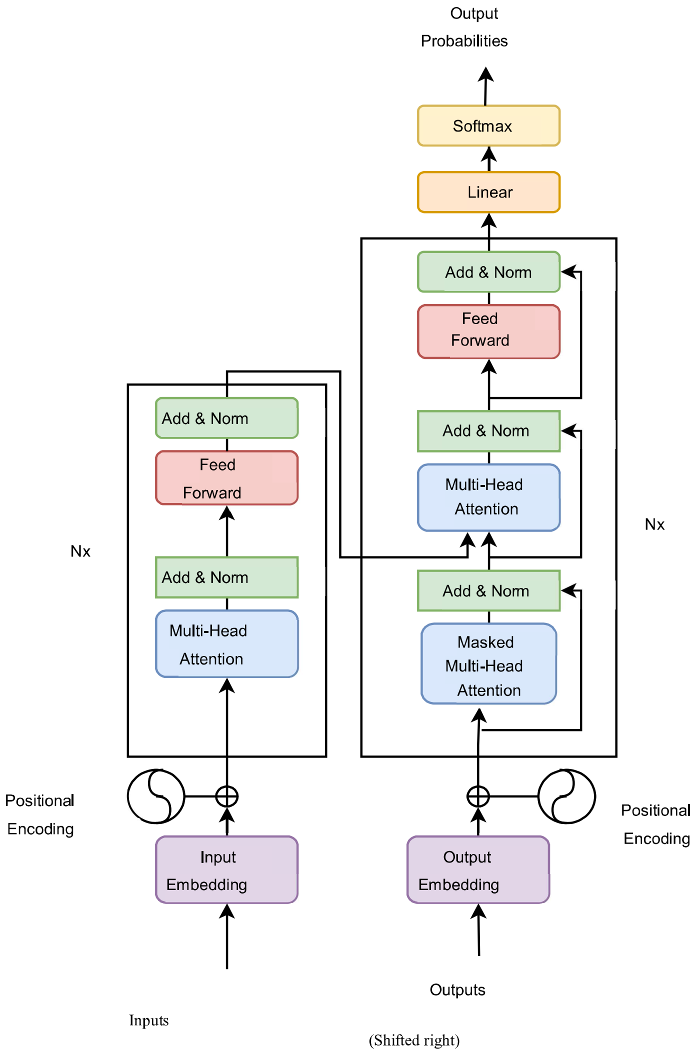
A sequence-to-sequence architecture, also known as a seq2seq architecture, is one of the ways that an encoder-decoder with attention may be used for time series forecasting. It has been demonstrated that this form of architecture is effective in a range of applications, such as machine translation and voice recognition. Additionally, it can be one approach to employ an encoder-decoder with attention to time series forecasting. Encoder and decoder in this architecture are commonly

RNNs or LSTMs, and the attention mechanism can be its own separate neural network if that is what the user desires.

When the links between the input features and the time series are complex, utilising encoder-decoder with attention mechanism in time series forecasting can lead to greater accuracy and performance overall. This is especially true in situations when the complexity of the relationships between the input features and the time series is high. This is especially true in situations in which the interactions between the input characteristics and the time series are highly complicated.

5.5 Transformer:

It has been demonstrated that transformers, a robust type of neural network model, are successful at a wide range of tasks that are associated with the processing of natural language. However, in order for them to be applicable in time series forecasting, they have been modified in such a way. The act of attempting to determine what the values of a collection of observations will be at some point in the future is referred to as the time series forecasting process.



vii. Figure 5.4: Transformer Structure

There are a few distinct approaches one may use when attempting to anticipate time series by utilising transformers. The problem can also be seen as a sequence-to-sequence learning problem, which is one of several possible approaches. In this method, the model is first given a collection of historical facts, and then, using those observations as a foundation, it formulates a set of projections for what the future may hold.

Before the transformer can be utilised for time series forecasting, its design will need to have a number of modifications made to it. To get started, the self-attention mechanism, which is a crucial part of the design of the transformer, is adjusted so that it can handle time series data. This is done so that the transformer can function properly. The transformer is now able to work correctly as a result of this adjustment. This is performed by affixing a positional encoding, which identifies where in time the observation occurs within the sequence, to each observation in the series. This allows for the achievement of the aforementioned goal. Second, the output of the transformer is altered in such a manner that rather than creating a single output value, it now creates a series of predicted values. This replaces the original behaviour of the transformer, which produced just a single output value. The functionality of typical transformer models is not designed to be like this at all.

For the aim of time series forecasting (TAM), several modifications of the transformer design have been proposed. Some examples of these modifications are the autoregressive transformer (AT), the encoder-decoder transformer (EDT), and the transformer autoregressive model. When applied to a range of time series forecasting assignments, it has been demonstrated that basic models are equally as successful as the most complicated ones in terms of accuracy.

5.6 Temporal Fusion Transformer:

The Temporal Fusion Transformer (TFT) is an architecture for deep learning that was built specifically for the goal of time series forecasting. In 2019, it was first described in a study authored by Bryan Lim and co-authors.

The Transformer architecture was initially developed for use in natural language processing-related activities. The TFT is a version of the Transformer architecture. A specific kind of neural network known as the Transformer architecture makes use of attention processes to determine the relative significance of the various components of the input sequence. In order for the Transformer design to be able to deal with time series data, the TFT alters the architecture by adding additional components that describe temporal interdependence.

The fundamental concept of TFT is to approach the challenge of predicting time series as if it were a sequence-to-sequence prediction issue. The TFT takes as its input a string of previously collected data, and produces, as its output, a string of forecasted future values. The TFT is made up of numerous levels of encoding and decoding, with each layer including its own unique combination of gating mechanisms, feed-forward neural networks, and multi-head self-attention.

The TFT generates accurate predictions by taking into account both the current time and the global context of the situation, in addition to using static information. Lagged variables are used to describe the temporal context, while a global context vector, which is a summary of the complete input sequence, is used to model the global context. Temporal context is modeled with the help of a global context vector. The model can also take into account static factors, such as the time of day or the day of the week, in order to give more information.

It has been demonstrated that the TFT is capable of performing effectively on a number of time series forecasting applications, such as forecasting the amount of electrical load and traffic. It has also been put to use in tasks involving categorization and the identification of anomalies. The TFT offers various benefits over traditional techniques of predicting time series, including its capability to manage non-linear and non-stationary data, as well as its capability to handle multiple seasonalities and long-term dependencies in the data. Other advantages include the capacity to handle non-linear and non-stationary data.

6.Result & Analysis

6.1 Performance Criterion:

Root Mean square error, Root mean absolute error, Root mean absolute scaled error, and coefficient of determination are the four metrics that have been used in our forecasting experiments to evaluate the efficacy of our models. RMSE stands for mean square error, RMAE stands for mean absolute error, RMASE stands for mean absolute scaled error and (R2) stands for coefficient of determination. Following are some equations that provide a mathematical representation of these metrics:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{6.1}$$

$$RMAE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|} \tag{6.2}$$

$$RMASE = \sqrt{\frac{MAE}{MAE_{naive}}} \tag{6.3}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2} \tag{6.4}$$

Here, y_i and ˆy_i represent the actual and predicted values, respectively, while ¯y_i indicates the mean of the actual values. MAE_{naive} is the MAE of a naive model that predicts the future value based on the present value.

6.2 Results

The multi-step solar irradiance is predicted by employing a number of different sequence-to-sequence attention-based models, and the data that is utilised to create the forecast originates from two different places. The data from the past 24 hours will be used as input for the model, which will then provide an estimate of the value of the Global Horizontal Irradiance (GHI) 12 hours ahead. Multi-step forward time series forecasting is the name given to this particular method of making predictions. Pytorch was utilised in the process of training and development.

The Transformer, GRU, and LSTM Encoder-Decoder models (GRU-ED and LSTM-ED), as well as the GRU and LSTM Encoder-Decoder models with attention (GRU-atn and LSTM-atn). The approaches that were utilised in the development of these models were discussed in greater detail in the part that came before this one. Pytorch's implementation in Pytorch Forecasting [27] was used to train the TFT model on the appropriate behaviour of its components. This was accomplished through the usage of Pytorch. Optuna [28] was utilised so that the various hyperparameters of the models could be adjusted.

This was important owing to the fact that hyperparameters like learning rate and hidden units have a substantial impact on how successfully a model performs. This is one of the reasons why this was necessary. The following is a list of the hyperparameters that have been selected for use in our modelling of forecasting and prediction.

Table iv: Parameters used for GRU-ED, LSTM-ED, GRU-attn and LSTM-attn

Parameter	GRU-ED	LSTM-ED	GRU-attn	LSTM-attn
Layers	1	1	1	1
Encoder hidden size	64	48	32	32
Decoder hidden size	64	48	32	32

Learning rate	0.0005	0.0005	0.0005	0.0005
Input sequence length	48	48	48	48
Teacher forcing ratio	0.6	0.5	0.6	0.5
Dropout	0	0	0	0
Batch size	256	256	256	256

Table v: Parameters used in TFT and Transformer

Transformer		**Temporal Fusion Transformer**	
Parameter	**Value**	**Parameter**	**Value**
Layers	3	Layers	1
Dmodel	24	Hidden size	32
Dff	16	Hidden continuous size	16
Attention heads	8	Attention heads	4
Learning rate	0.0005	Learning rate	0.0001
Input sequence length	48	Input sequence length	48
Dropout	**0.2**	**Dropout**	**0.2**
Batch size	256	Batch size	256

Both the GRU and LSTM layers, which are utilised in the encoding process of the attention-based model, are versatile and able to operate in either direction. The decoder in the model takes into consideration the previous value of the target before coming to a conclusion about the future. During training, there are a number of different strategies that may be utilised to make an attempt at anticipating how the decoder would carry out its function. One possible strategy that might be utilised is known as a recursive prediction. In other words, the outputs of the decoder that have been anticipated in the past flow back into the decoder until an output has been created that has the requisite length.

If the predictions made at the beginning of the training process are too erroneous, the errors will compound during the length of the sequence, making it more challenging for the model to learn and fast converge on a solution. One of the downsides of using this method is that it does this. The use of force or intimidation by the instructor is yet another strategy [29, 30]. The decoder of the model will, at various points in the process of teacher forcing, provide predictions based on the actual value of the objective that came before it. It makes certain that the model of the series continues to bear a significant resemblance to the real sequence. There is a single shortcoming associated with this strategy, and that is the fact that during the inference phase, there is no true target value. The process of inference is different from training in the sense that, during the latter, you are forced to repeatedly make predictions. Training does not include this need.

Therefore, the next strategy that we will employ is to combine the two tactics that we have already tried. After that, the two methods will be combined through the use of a ratio. In certain situations, the actual number will be sent to the decoder, while at other times, the predicted value will be used in the construction of the decoder instead. This specific number is what the TFR is referring to. Within the context of this specific experiment, the Adam optimizer was utilised in order to ascertain which response was the most suitable.

In addition to the fundamental MLP and Naive models, the efficiency of the sequence-to-sequence models is also assessed and contrasted here. This is done both individually and comparison is done with the other models. The Naive model extrapolates from the most recent value or period in order to generate predictions about future values or periods. These extrapolations are based on the most recent value or period. The naïve model will base its estimate of the irradiance that will be received on the next day on the value that was recorded the day before. In addition to this, a straightforward MLP model has been developed that is capable of producing recursive sequence predictions in order to facilitate the evaluation of both of our sequence models.

The MLP model might be effective at times [31, 32] when one is aiming to create accurate predictions regarding time series. During the course of this investigation, an MLP model was utilised to analyse the data. It is made up of two secret levels, and on each of those levels, there are sixty-four components that are concealed.

Measurements that were used in the evaluation of these forecasting models for the two distinct locations are shown in Table 6.3 and 6.4. The most useful findings are shown in bold text.

Table vi: Forecasting metrics for Dhaka

Model	Dhaka			
	RMSE	RMAE	RMASE	R^2
Naive	0.549545267	0.531977443		0.788669766
MLP	0.424264069	0.492950302	0.926282894	0.880340843
GRU-ED	0.423083916	0.481663783	0.904986188	0.880908622
LSTM-ED	0.427784993	0.485798312	0.913236005	0.877496439
GRU-attn	0.391152144	0.480624594	0.903327183	0.899444273
LSTM-attn	0.4	0.467974358	0.879204186	0.893867999
Transformer	0.441021541	0.520576603	0.978263768	0.870057469
TFT	0.392428337	0.463680925	0.871206061	0.897775027

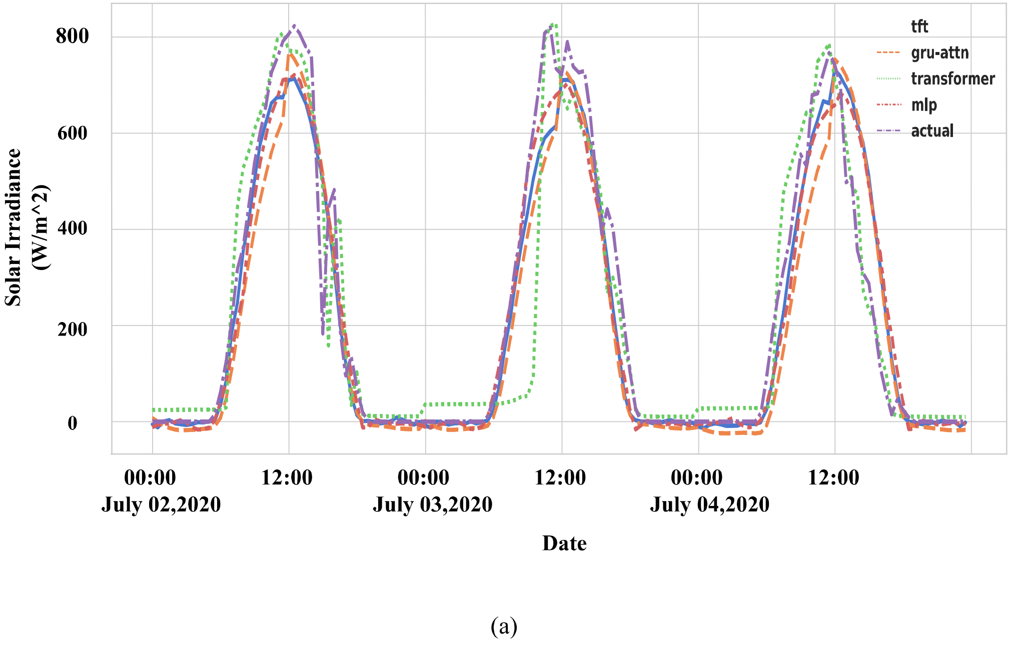
Table vii: Forecasting metrics for Cox’s Bazar

Model	Cox’s Bazar			
	RMSE	RMAE	RMASE	R^2
Naive	0.526307895	0.512835256		0.81731267
MLP	0.413521463	0.490917508	0.957078889	0.892188321
GRU-ED	0.389871774	0.467974358	0.912688337	0.904433524
LSTM-ED	0.394968353	0.47644517	0.928977933	0.902219485
GRU-attn	0.4	0.491934955	0.959166305	0.899444273
LSTM-attn	0.404969135	0.485798312	0.947100839	0.896660471
Transformer	0.431856458	0.54405882	1.060660172	0.881476035
TFT	0.38340579	0.458257569	0.893308457	0.907744457

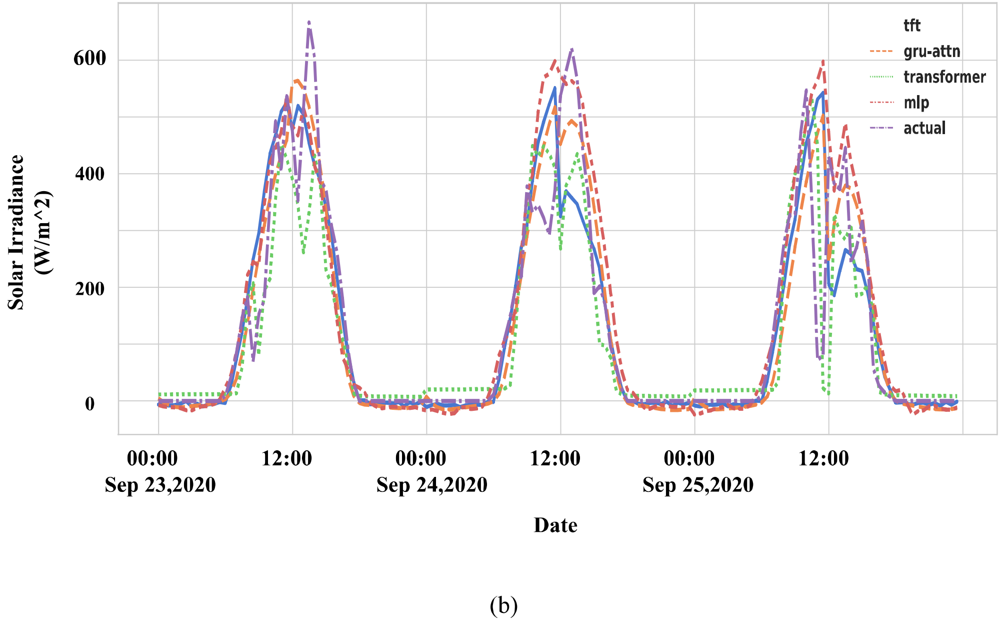
When compared to the "naive" model, almost all of the forecasting models that are now in use are able to provide predictions that are far more accurate than those generated by the "naive" model. This is seen in tables 6.3 and 6.4. The table also reveals that TFT performs better than the other models in both locations for the most of the metrics that are being compared. This is the case for the majority of the metrics. The ANN model comes in second place, after the Transformer model, as the one that has the second-worst performance overall when compared to the other models. The Naive model comes in first.

Because sequential models include recurrent structures that are able to store sequential data, sequential models often perform better than MLP models when it comes to making predictions regarding time series. This is because MLP models do not have such structures. In Cox's Bazar, the performance of both GRU-ED and LSTM-ED is much superior to that of MLP across the board. GRU-ED has shown to be the most successful. In Dhaka, the performance of MLP is superior to that of LSTM-ED in terms of the RMSE and RMASE values, but the performance of LSTMED is superior in terms of the RMAE and R2 values. The results achieved with GRU-ED are superior to those obtained with MLP and LSTM-ED, even when this situation is taken into consideration.

In Dhaka, the GRU and LSTM attention models perform better than the MLP and encoder decoder models, while the GRU-attn model performs the best overall and even performs better than TFT in terms of the RMSE and R2 score. In general, the GRU-attn model performs better than the TFT model. The RMSE and RMAE values of 0.389871774 and 0.467974358 for the GRU-ED model and 0.404969135 and 0.485798312 for the attention models, respectively, reveal that the GRU-ED model operates more successfully in Cox's Bazar than the attention models do.



(a)

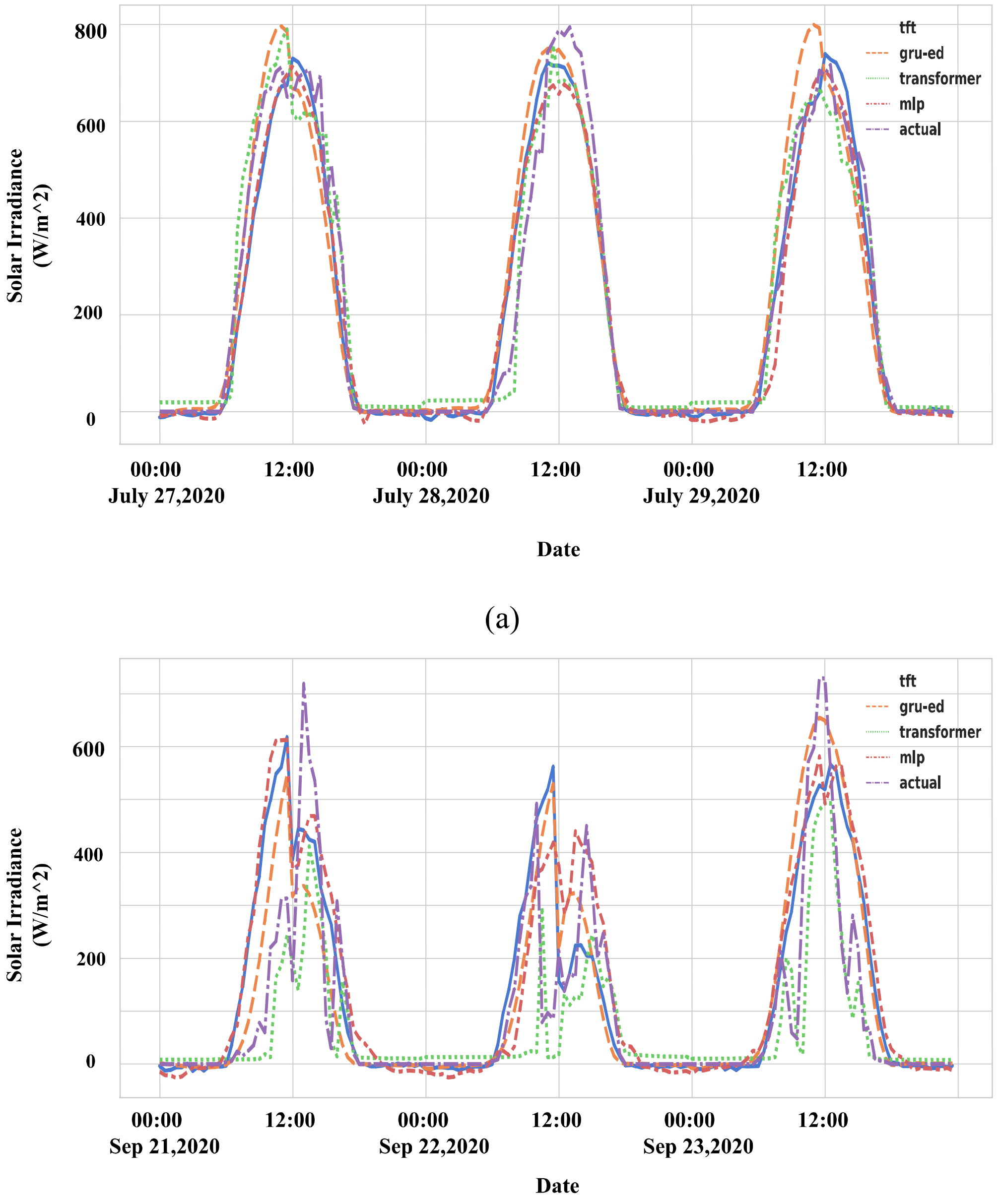


(b)

viii. Figure 6.1: Predicted solar irradiance for different models in Dhaka during (a) clear-sky (b) cloudy days

The performance of the Naive model is a little bit better than that of the Transformer model, which has the worst results in both categories. The Transformer model did not perform as well when it was put to the test, despite the fact that it had a good performance throughout the time that was committed to training. In conclusion, among all the models that were discovered in Cox's Bazar, the R2 value of the TFT model is the highest, and its RMSE, RMAE, and RMASE loss are the lowest. In addition, the TFT model has the lowest RMASE loss. GRU-attn is the only model that outperforms TFT in Dhaka. Its RMSE and R2 values are 0.391152144 and 0.899444273 respectively. Both the RMAE and RMASE scores for TFT are the highest possible in this specific domain. The TFT model is able to incorporate a wide range of different forms of data, such as static covariates, inputs that are known for at least the foreseeable future, and temporal variables that are only known up to the present instant in time. The model might potentially be trained with the data from a variety of time series. In order for this strategy to be successful in accomplishing its objectives, it utilises a temporal self-attention decoder in conjunction with a multi-head attention mechanism.

When this is analysed, it offers further information on the relevance of features, which, in turn, makes it feasible to detect long-term dependencies. Figures 6.1 and 6.2, respectively, illustrate both the actual facts as well as what the various models predict will take place at both places and in both types of weather. The accuracy of the data is 24 times lower than the prediction capacity of the algorithms that we have developed. When there are a lot of clouds in the sky, like the day that is seen in the image, algorithms are able to show how inaccurate and unpredictable the data that is collected from the sun may be. Cloudy days make it difficult to predict what the weather condition will be, therefore clear skies allow for more effective performance from weather forecasting models.



(a)

(b)

ix. Figure 6.2: Predicted solar irradiance for different models in Cox’s Bazar during (a) clear sky (b) cloudy days

Individuals who live in Cox's Bazar have a higher proportion of success overall when it comes to generating correct estimates, in comparison to persons who live in other places. Almost every single model for making forecasts does exceptionally well in this region, outperforming its peers in other parts of the world. This may be due to the fact that the seasonal pattern is more constant in this region and that there is less residue or unpredictability as a consequence of the varied and overcast weather. Alternatively, this might be because the seasonal pattern is more consistent in this region due to the fact that the climate is more stable. There is always the potential that the ever-changing weather is another reason why there is less residue or unpredictability. Considering how frequently the weather shifts, there is always this option. It has been able to see the same information when the Naive model is employed; however, the error values are far smaller in Cox's Bazar than they are in Dhaka.

If the Naive model is used to forecast the future period based on the preceding period, then there is a greater likelihood that our assumption that the data from Cox's Bazar follow seasonality is right. This is because the Naive model uses just the most basic information. This is because the naïve model assumes that the following era would be precisely the same as the period that came before it, therefore this result is not surprising. The RMSE values for the TFT model are 0.392428337 and 0.38340579, while the RMAE values are, respectively, 0.463680925 and 0.458257569 at both locations. The numbers for RMSE are smaller than the values for RMAE. This suggests that the TFT model has a higher degree of coherence all around. Also, although attention models are successful in both cities, the success rate of these models is significantly higher in Dhaka for some unknown reason. This illustrates that the TFT model possesses a degree of stability that is superior to that of other models. Other models lack the degree of stability that the TFT model possesses.

After analysing the data, the TFT model was shown to be capable of producing estimations of the irradiation coming from the sun that was quite accurate. TFT performed noticeably better than the other varieties in every metric that was considered relevant to the evaluation of performance. Following the TFT, the attention-based GRU Encoder-Decoder turned out to be the method that performed the best. In addition to this, it was given the MSE and R2 values that were regarded as the highest possible of any institution in the Dhaka area.

Table viii: Overall Performance Metrics

Model	RMSE	RMAE	RMASE	R2
Naive	0.538516481	0.522494019		0.803741252
MLP	0.419523539	0.491934955	0.94127573	0.886002257
GRU-ED	0.40620192	0.475394573	0.909945053	0.893308457
LSTM-ED	0.411096096	0.480624594	0.919782583	0.891066776
GRU-attn	0.396232255	0.485798312	0.929516003	0.898888202
LSTM-attn	0.402492236	0.47644517	0.911592014	0.895544527
Transformer	0.435889894	0.519615242	0.994484791	0.875785362
TFT	0.388587185	0.460434577	0.880908622	0.902773504

The Transformer model that was utilised for the Time Series had the very poorest performance of any of the various versions that were used. This model was selected since it was the only one available at the time. Findings provided by the TFT and attention model are also more reliable and consistent in both locations, however, the predictions made by the other models for the two different sites do not always match with one another. This is the case despite the fact that the TFT and attention model are more trustworthy.

It is absolutely necessary to keep in mind that the TFT and attention models require additional data in order to provide reliable predictions. Due to the fact that the model is limited and intricate, it is essential to keep this in mind at all times. Despite the fact that the TFT is a complex model that has to be run for a longer length of time in order to give results, these findings are quite accurate and can be applied to sophisticated multivariate time series data. This is the case even though these findings can be applied. In addition to that, the TFT is a model that needs to be carried out. In addition, the performance of this model is noticeably higher than that of earlier statistical models, while at the same time requiring a noticeably decreased quantity of processing of the data. This is because the model requires a considerably reduced amount of the original data to be processed. TFT is beneficial because it has the ability to identify small temporal correlations as well as long-term connections, both of which can be rather complex. These are two of the many reasons why Tapping Freedom Technique (TFT) is so beneficial. This model is adaptable enough to be utilised in a broad variety of additional difficult time series applications due to its high level of flexibility.

As can be seen in Table 6.5, the overall performance of the test dataset when compared to a variety of alternative models is also taken into consideration. The outcomes of the trials indicate that TFT achieves a superior value in all of the error measurements, with an RMSE value of 0.38340579 and an RMAE value of 0.458257569, respectively. This is the conclusion that can be drawn from the findings of the investigations, and it can be obtained from those findings using the outcomes of the investigations. Following the completion of the TFT, it was discovered that the performance of Encoder-Decoder models was inferior to that of Attention models. Information may be extracted from extremely lengthy sequences of input data by using attention-based models, which have this capability.

On the other hand, the Encoder-Decoder architecture is unable to gather any information since the length of the context vector representation that it employs is already set in stone. This prevents it from being able to collect any data. This is a striking departure from the design that was discussed before. GRU and LSTM have both been proven to operate in a manner that is equivalent to one another, with GRU performing somewhat better than LSTM in the Encoder-Decoder and Attention models. This has been demonstrated. In spite of the fact that their architectures are different, it has been demonstrated that the GRU and the LSTM both operate in a manner that is comparable to one another.

7.Conclusion & Future Work:

As part of the scope of this study, an Attention-based deep learning framework has been presented for the multivariate multistep Time Series Forecasting issue. This study aimed to investigate the relationship between attention and deep learning. The attention-based encoder-decoder model, the transformer model, and the Temporal Fusion Transformer (TFT) model are all put to the test in Bangladesh in order to provide accurate estimates of sun irradiance at two separate locations in the country. The components of the dataset that have been gathered include the cloud cover, various aspects of the weather, and the historical values of solar irradiance, among other things.

Due to the fact that it is notoriously difficult to produce accurate weather predictions, it is impossible to exactly determine the amount of irradiation that the sun emits. This results in instabilities within the grid, which is made up of a variety of components, all of which operate in tandem with one another. To discover whether or not the attention models are capable of more accurate prediction of the future than the models that are currently being used for this purpose, the primary goal of this work is to determine whether or not the attention models are capable of doing so. These algorithms are able to learn about both long-term and short-term dependencies, which allows them to perform more effectively and produce better outcomes.

In addition to this, they perform their functions efficiently to produce excellent results. These algorithms are also evaluated in contrast to many other models, including LSTM, GRU Sequence, MLP, and Naive, all of which have been shown to be helpful in the process of developing correct predictions. The findings obtained from these comparisons are utilised as the basis for drawing judgments on the efficiency of the algorithms.

8.References

[1] Ehsanul Kabir, Pawan Kumar, Sandeep Kumar, Adedeji A Adelodun, and Ki-Hyun Kim. Solar energy: Potential and future prospects. *Renewable and Sustainable Energy Reviews*, 82:894–900, 2018.

[2] Dolf Gielen, Francisco Boshell, Deger Saygin, Morgan D Bazilian, Nicholas Wagner, and Ricardo Gorini. The role of renewable energy in the global energy transformation. *Energy strategy reviews*, 24:38–50, 2019.

[3] Amir Shahsavari and Morteza Akbari. Potential of solar energy in developing countries for reducing energy-related emissions. *Renewable and Sustainable Energy Reviews*, 90: 275–291, 2018.

[4] Seyed Ehsan Hosseini and Mazlan Abdul Wahid. Hydrogen from solar energy, a clean energy carrier from a sustainable source of energy. *International Journal of Energy Research*, 44(6):4110–4131, 2020.

[5] Benjamin Huybrechts. Social enterprise, social innovation and alternative economies: Insights from fair trade and renewable energy. *Alternative economies and spaces. New perspectives for a sustainable economy*, pages 113–130, 2013.

[6] Vikas Khare, Savita Nema, and Prashant Baredar. Solar–wind hybrid renewable energy system: A review. *Renewable and Sustainable Energy Reviews*, 58:23–33, 2016.

[7] Mohamed Abdel-Nasser and Karar Mahmoud. Accurate photovoltaic power forecasting models using deep lstm-rnn. *Neural Computing and Applications*, 31:2727–2740, 2019.

[8] A Karthick, K Kalidasa Murugavel, Aritra Ghosh, K Sudhakar, and P Ramanan. Investigation of a binary eutectic mixture of phase change material for building integrated photovoltaic (bipv) system. *Solar Energy Materials and Solar Cells*, 207:110360, 2020.

[9] Abdelhakim Mesloub, Artira Ghosh, Ghazy Abdullah Albaqawy, Emad Noaime, and Badr M Alsolami. Energy and daylighting evaluation of integrated semitransparent photovoltaic windows with internal light shelves in open-office buildings. *Advances in Civil Engineering*, 2020:1–21, 2020.

[10] Maria Khalid, Katie Shanks, Aritra Ghosh, Asif Tahir, Senthilarasu Sundaram, and Tapas Kumar Mallick. Temperature regulation of concentrating photovoltaic window using argon gas and polymer dispersed liquid crystal films. *Renewable Energy*, 164: 96–108, 2021.

[11] Saad Mekhilef, Rahman Saidur, and Masoud Kamalisarvestani. Effect of dust, humidity and air velocity on efficiency of photovoltaic cells. *Renewable and sustainable energy reviews*, 16(5):2920–2925, 2012.

[12] Jessica Wojtkiewicz, Matin Hosseini, Raju Gottumukkala, and Terrence Lynn Chambers. Hour-ahead solar irradiance forecasting using multivariate gated recurrent units. *Energies*, 12(21):4055, 2019.

[13] Walter Nsengiyumva, Shi Guo Chen, Lihua Hu, and Xueyong Chen. Recent advancements and challenges in solar tracking systems (sts): A review. *Renewable and Sustainable Energy Reviews*, 81:250–279, 2018.

[14] Ramanan Pichandi, Kalidasa Murugavel Kulandaivelu, Karthick Alagar, Hari Kishan Dhevaguru, and Suriyanarayanan Ganesamoorthy. Performance enhancement of photovoltaic module by integrating eutectic inorganic phase

change material. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, pages 1–18, 2020.

[15] Jan Kleissl. *Solar energy forecasting and resource assessment*. Academic Press, 2013.

[16] Ronald B Melton, Kevin P Schneider, Eric Lightner, Thomas E Mcdermott, Poorva Sharma, Yingchen Zhang, Fei Ding, Subramanian Vadari, Robin Podmore, Anamika Dubey, et al. Leveraging standards to create an open platform for the development of advanced distribution applications. *IEEE Access*, 6:37361–37370, 2018.

[17] Emre Akarslan, Fatih Onur Hocaoglu, and Rifat Edizkan. Novel short term solar irradiance forecasting models. *Renewable Energy*, 123:58–66, 2018.

[18] Benedikt Schulz, Mehrez El Ayari, Sebastian Lerch, and S´andor Baran. Post-processing numerical weather prediction ensembles for probabilistic solar irradiance forecasting. *Solar Energy*, 220:1016–1031, 2021.

[19] Dazhi Yang. A universal benchmarking method for probabilistic solar irradiance forecasting. *Solar Energy*, 184:410–416, 2019.

[20] Vincent Le Guen and Nicolas Thome. A deep physical model for solar irradiance forecasting with fisheye images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 630–631, 2020.

[21] Liang Ye, Zhiguo Cao, Yang Xiao, and Zhibiao Yang. Supervised fine-grained cloud detection and recognition in whole-sky images. *IEEE Transactions on Geoscience and Remote Sensing*, 57(10):7972–7985, 2019.

[22] Huaizhi Wang, Ren Cai, Bin Zhou, Saddam Aziz, Bin Qin, Nikolai Voropai, Lingxiao Gan, and Evgeny Barakhtenko. Solar irradiance forecasting based on direct explainable neural network. *Energy Conversion and Management*, 226:113487, 2020.

[23] Manal Marzouq, Hakim El Fadili, Khalid Zenkouar, Zakia Lakhliai, and Mohammed Amouzg. Short term solar irradiance forecasting via a novel evolutionary multi-model framework and performance assessment for sites with no solar irradiance data. *Renewable Energy*, 157:214–231, 2020.

[24] Pratima Kumari and Durga Toshniwal. Long short term memory–convolutional neural network based deep hybrid approach for solar irradiance forecasting. *Applied Energy*, 295:117061, 2021.

[25] Bixuan Gao, Xiaoqiao Huang, Junsheng Shi, Yonghang Tai, and Jun Zhang. Hourly forecasting of solar irradiance based on ceemdan and multi-strategy cnn-lstm neural networks. *Renewable Energy*, 162:1665–1683, 2020.

[26] Xiaoqiao Huang, Qiong Li, Yonghang Tai, Zaiqing Chen, Jun Zhang, Junsheng Shi, Bixuan Gao, and Wuming Liu. Hybrid deep neural model for hourly solar irradiance forecasting. *Renewable Energy*, 171:1041–1060, 2021.

[27] Pytorch forecasting documentation—pytorch-forecasting documentation. https://pytorchforecasting.readthedocs.io/en/stable/index.html. (Accessed on 03/21/2023).

[28] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[29] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[30] Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. *Advances in neural information processing systems*, 29, 2016.

[31] Tianping Zhang, Yizhuo Zhang, Wei Cao, Jiang Bian, Xiaohan Yi, Shun Zheng, and Jian Li. Less is more: Fast multivariate time series forecasting with light samplingoriented mlp structures. *arXiv preprint arXiv:2207.01186*, 2022.

[32] Pedro Henrique Borghi, Oleksandr Zakordonets, and Jo˜ao Paulo Teixeira. A covid19 time series forecasting model based on mlp ann. *Procedia Computer Science*, 181: 940–947, 2021.