National College of Ireland

Forecasting Energy Generation in Spain from Renewable Sources Using Time Series and Neural Network Models

MSc in Data Analytics

Saranya Varshni Roshan Karthikha
Student ID: x20154801

School of Computing
National College of Ireland

Supervisor: Dr. Bharathi Chakravarthi

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Saranya Varshni Roshan Karthikha
Student ID:	x20154801
Programme:	MSc in Data Analytics
Year:	2021
Module:	MSc in Data Analytics
Supervisor:	Dr. Bharathi Chakravarthi
Submission Due Date:	16/12/2021
Project Title:	Forecasting Energy Generation in Spain from Renewable Sources Using Time Series and Neural Network Models
Word Count:	5570
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Saranya Varshni Roshan Karthikha
Date:	16th December 2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Forecasting Energy Generation in Spain from Renewable Sources Using Time Series and Neural Network Models

Saranya Varshni Roshan Karthikha
x20154801

Abstract

Consumption of power has become an inevitable part of one's being. With the constant development in economy and population, the energy demand never falls down instead it's exponentially raising. In recent decades, the investments made in products and industrial sectors paves the path for 24/7 energy demand. It can be strongly stated that economic development in a country is highly proportional to energy generation. On a wide scale, power generation sources can be categorized as renewable energy sources (i.e.) solar, wind, hydro, geothermal, etc and fossil fuels which are non-renewable energy sources. Combustion of such non-renewable sources will lead to emission of toxic gases and CO2 into the environment leading to harmful consequences and these non-renewable sources get deprecate soon. So, it is safe to rely on the natural source of power generation and it is safe for the atmosphere as well. However, there is a lot of uncertainty factor associated with power generation from such environment-friendly sources. They are highly dependent on the climate and forecasting such challenging features could be worthy of working on. This energy generation forecasting will help many private and governmental organizations to develop a balance between the supply and demand chain. Hence, forecasting various energy sources using different neural network models and the ARIMA time series model will constitute a great research study as well as benefit the stakeholders from the power production sector and help them to be aware of the future trends before any investment plans.

1 Introduction

In the present decade, everything around works upon electricity consumption, and the demand for electricity in every developing country raises like peaks. Economic advancements and GDP rates are highly interrelated with the total energy produced and used for the improvements of the country. Electricity could be generated from different sources. In this research, renewable energy sources (solar, wind, hydro, biomass, etc) are being considered over non-renewable sources like fossil fuels. It is highly recommended to use such natural energy sources as they exhibit very less carbon while used as fuel for electricity generation. Unlike fossil fuels, these sources never get depleted. Hence, this study focuses on forecasting power generation from various renewable energy generation sources like a solar, hydro water reservoir, biomass, fossil hard coal, and nuclear. An intense literature review was done to understand the existing research activities and latest advancements.

The advantage of performing this research is to identify the best model that is with the best possible accuracy for long-term forecasting energy generation. This study also guides with the most promising energy source for the future. Energy generated from different sources is integrated into power grids, hence the forecast will help in the efficient power integration and storage manner.

Research question:

Whether historical electricity generation data can accurately forecast future electricity generation in the long term using LSTM and ARIMA models?

A higher level of accuracy in the prediction model can help the different policymakers in spotting areas of unwanted energy generation and can be avoided.

The following research objectives were followed:

- To perform a literature review on recent research activities in the domain
- Research on data availability and selective attributes
- Decide on the technology involved with the research and perform prerequisite framework implementations
- Implement the ARIMA, LSTM, LSTM - CNN, Stacked LSTM models
- Evaluate and interpret the results by visualizing them
- Critically analyze the implementation and discuss the future works and refinements

Following this introduction, Section 2 showcases works that have already been conducted in the domain of energy generation and other domains where similar algorithms have been utilized. In Section 3, I propose the methodology being deployed in order to model the aggregated data. Section 4 will look at the models being employed and contain a description of each. Section 5 will focus on the implementation of the algorithms. The following section will examine the results of the experiment and discuss the implications of the results for the research (section 6).

2 Related Work

The application of deep-learning models and arithmetic algorithms in the power generation utility has revolutionized many research groups. The application of Artificial intelligence, leading to the advancements in power grid technologies has paved a new path for researchers in the field of renewable and non-renewable energy sources. With the development of different countries, the renewable energy generation investment increases sustainably with records to show reduced greenhouse gas emission (Paul et al.; 2021).

A predictive study that incorporates all the use of renewable energy sources, urbanization, fossil fuel, and wide economic growth is required to showcase the dynamics of energy consumption. This examines the CO2 emissions from 1990 till 2020 in China. Autoregressive Distributed Lag (ARDL) model is being used to test the long short-term energy expenditure. The observations suggest that there is a strong relationship between

the GDP per capita and CO2 emission. Every 1% increase in fossil consumption raises the CO2 emission by 0.235% in a long run. However, for every 1% increase in usage of renewable energy, CO2 emission decreases by 0.259% (Li and Haneklaus; 2021).

(Bhatia et al.; 2021) proposed an ensemble for forecasting short-term electricity price prediction. Performed a few feature selection and exploratory data analyses of renewable energy. Every methodology was compared with six other states of art models for the next one hour of forecasting. This ensemble model proved to be valid and performed higher with accuracy where MAE was 1.38. With these results, the model is more suitable for training online with the lowest computation timing of 136 seconds. The ensemble model blends the boosting and bagging merits while in the stacking frame and further by reducing the variance that got added due to the stacking process with the addition of bootstrap aggregation.

(Ruhnau et al.; 2020) argues that recent studies perform forecasting of data points using different statistical and intelligence techniques or a hybrid model of both. The evaluation strongly focuses on the accuracy of the models. However, the relative performance of every methodology varies with different accuracy measures. This study also focuses on the asymmetry effect which occurs due to different market conditions. Some biased electricity forecasts can be carried out with probabilistic density. The forecasting models utilized are the clear sky model, linear model, and artificial neural network. With this research, it was concluded that not just the accuracy and asymmetry of a model, but also the correlation has a significant impact on the forecast. However, it was difficult to derive results as the distribution of forecast error and price spread.

2.1 Research related to statistical time series forecasting

2.1.1 ARIMA:

A comparative study was performed by (Natarajan and Karatampati; 2019) on reviewing various forecasting techniques for solar and wind renewable energy. Statistical models like ARIMA, SMA, ES, ARMA, and neural network models like LSTM, RNN are being experimented with. Both statistical and neural models were considered good for any short-term predictions. While the error rate increases when the forecasting time period increases. The nonlinearity issue was effectively addressed by the ANN model. The long sequence dependency issue that was raised in RNN was resolved by LSTM implementation. It was finally concluded that the LSTM model was most suitable for short-term and mid-term forecasting.

(Bantupalli and Matam; 2017) also used ARIMA and ANN to forecast the wind energy generation which was based on Empirical Mode Decomposition (EMD). EMD is used to decompose data into Intrinsic Mode Functions(IMF) and other residuals. ANN was implemented with 100 hidden nodes and the hidden nodes that gave good MAPE were recorded and the nodes corresponding with the weights were used for other hidden nodes with lesser performance. ARIMA model was observed to perform better than ANN with 5.609% of MAPE.

Another statistical approach for the prediction of short-term generation from renewable sources was experimented by (Nair et al.; 2021) on economic dispatch and unit commitment optimization. Methodologies like ARIMA SVM, GAM, and other hybrid models were executed. In this study, only 29 days of data were considered as historic data and the prediction was made for 6 hours. Each forecast uses a rolling window approach, where 19 hours of data is input and the next 6 hours is the output. ARIMA

being the statistical model performs well for some iterations but at one point seemed overfitting. A hybrid model created with GLM-SVM outperformed other models with the highest accuracies.

(Meenal et al.; 2018) conducted research on solar energy potential forecasting using (RF) Random Forest and learning algorithms. This RF model helps in creating the solar potential map of India. Total forecasted global solar radiation ranged between 13 and 21 MJ/m square per day. These results indicate good solar potential. The results showcase that, RF model performs the best with the accuracy level. The forecasted values are checked with original GSR data and it was implemented using WEKA software.

2.2 Research based on Neural Network models

2.2.1 LSTM

(Gencer and Başçiftçi; 2021) built a predictive model for forecasting android vulnerabilities that are time-dependent. The model estimates the total number of vulnerabilities available and risk evaluation. Various deep learning models like CNN-LSTM, ConvoLSTM, multilayer perceptron, CNN, LSTM were deployed to forecast and the models which had the lowest loss and error rate were selected as the best forecasting model. The results revealed that the ARIMA model produced an error rate of 18.449 whereas, LSTM has 26.830 as its error rate. It was concluded that learning models like LSTM can produce error rates that as close to the statistical time series models in spite of fewer data points.

2.2.2 LSTM-CNN

:

(Kumari and Toshniwal; 2021) combined LSTM and CNN to form a new hybrid model for hourly global horizontal irradiance prediction. LSTM was used to extract time-series features from the data. Further CNN was used, which is used to extract the spatial features from the heat map(i.e.) correlation coefficient values. The data time span considered was for a year and four seasons. The observations from this experiment suggest that the hybrid LSTM-CNN model proposed is a great alternative to any of the short-term predictions as the prediction accuracies are high even with limited data points.

2.2.3 Stacked LSTM

(Cui et al.; 2020) used an RNN based deep learning model for short-term prediction of traffic. The author proposed a unidirectional and bidirectional LSTM to assist the RNN structure for forecasting. The bidirectional LSTM is one key component that is used to record the forward and backward time series dependencies in the spatiotemporal datasets. Data imputing has become an unavoidable step in data cleaning. The author also proposes a data manipulation mechanism in LSTM by introducing an imputing unit to manipulate the missing values. The bidirectional LSTM is incorporated in the unidirectional LSTM workflow. The results depict that the single or bidirectional LSTM, especially the BDLSTM model have recorded the highest performance with regards to robustness and accuracy.

(Jin et al.; 2022) discussed the existing forecasting system for renewable energy consumption and its shortcomings. The author proposed a novel model (hybrid model) that

combines the (SSA) Singular Spectrum Analysis and parallel LSTM. Further, the decomposition using SSA improved the performance of the hybrid model. SSA extracts the attributes as sub-signals and removes the noise in data. These sub-signals are later taken up by each neural network for prediction. The parallel LSTM trained the LSTM network concurrently and combines the predicted outputs to form the finalized results. With the result observations, the proposed hybrid model exhibited great prediction performance.

Another deep learning approach in predicting the solar energy generation using a stacked ensemble algorithm (DSE-XGB) that combines two learning algorithms named ANN and LSTM was proposed (Khan et al.; 2021). The results from these combined models are passed through the extreme gradient boosting algorithm to increase the performance. The hybrid model's output was then examined comparatively with the individual forecast results of ANN, Bagging, and LSTM. The proposed hybrid model (i.e.) DSE-XGB model shows the best combination for stability even with higher variance with weather data and exhibits enhanced R squared of 10 percent –12 percent when compared to other models.

(ArunKumar et al.; 2021) conducted research on predicting the global impact of COVID-19 using a Recurrent neural network. The Gated Recurrent Units and LSTM with RNN were deployed to predict the future patterns of the COVID-19. The forecast was made for 60 days with RNN-GRU and RNN-LSTM models. The evaluation metrics were considered to be RMSE and MSE. Both the models performed best for a different set of countries. There were some inconsistencies observed in the data that was reported.

To conclude, the majority of the study is aimed on forecasting future events and measures. Different procedures like the creation of a hybrid model for the improvement of accuracy level. Most of the related work summarized in ARIMA being the best model in case of a staistical model. Also in few research articles, Neural network models like LSTM outperformed the classic time series models. It is always worth forecasting the upcoming values, which guides us in further preparation and to draw precise measures.

3 Methodology

3.1 Introduction

This section includes the overall process of research activities and the flow of the project. The procedure of extracting insightful information from the raw data source for making decisions is defined as Data Mining (Yahya and Osman; 2019). This research follows a tailored methodology or an enhanced version of CRISP-DM that incorporates all the research requirements for the process of data mining. The generic business methodology of CRISP-DM is modified as shown in Figure 1.

Stage 1 is the research area identification and the fundamental analysis of existing research establishments and strong research aim or research question composition. Followed by the next critical stage, dataset identification and preprocessing and transformation of the same. The third stage is modeling the neural network and time series models. Further, the next stage with an evaluation of different model performances. Finally, the visualization of the results and documentation of the same could be considered as the deployment of the whole methodology.

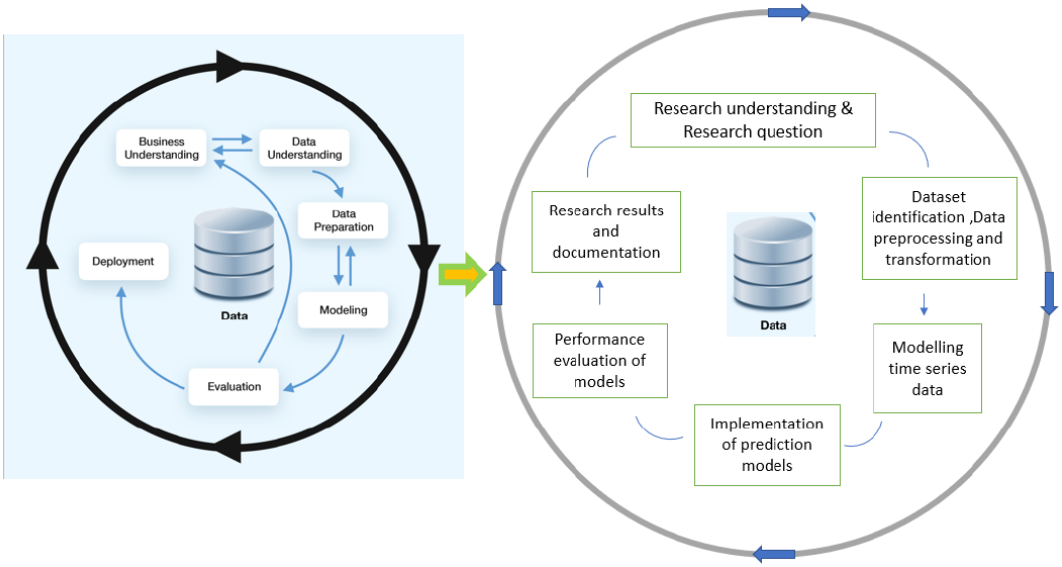


Figure 1: Methodology Stages

3.2 Architectural Design

To form a robust and efficient project structure, architecture is designed with 3 different layers. These layers include the data persistent layer that comprises data source, data extraction, data clean, load, and transformation. This also specifies the environment and tools used as a part of data preprocessing activity. The second layer is the business logic layer, where the transformed data is used in different neural network models like LSTM and time series models like ARIMA. Different evaluation parameters are considered to gauge the model performance. Finally, the results are visualized and documented as the outcome of the experiment or research for the client-side. Figure 2 represents the architectural design.

3.3 Data Source

For this research, the dataset is sourced from a public open-source data repository named Kaggle. It counts to have 29 features and 34065 energy consumption records in Spain from various renewable and non-renewable energy sources. These data points were extracted from (TSO) Transmission Service Operator, ENTSOE which is an open portal. These records specify the amount of energy generation and consumption units. There is a 'time' column that contains the hourly entries of each consumption rate that spans across 4 years of data from 2014 to 2018. Using this time as index different energy sources like biomass, solar, hydro, nuclear, and fossil consumption are forecasted.

3.4 Data Aggregation

The initial step with data was to check on the granularity of the dataset. It was observed to be an hourly granularity. The primary data processing and implementation required Jupyter environment created by Anaconda. With the help of Pandas library, the CSV file was imported into a data frame.

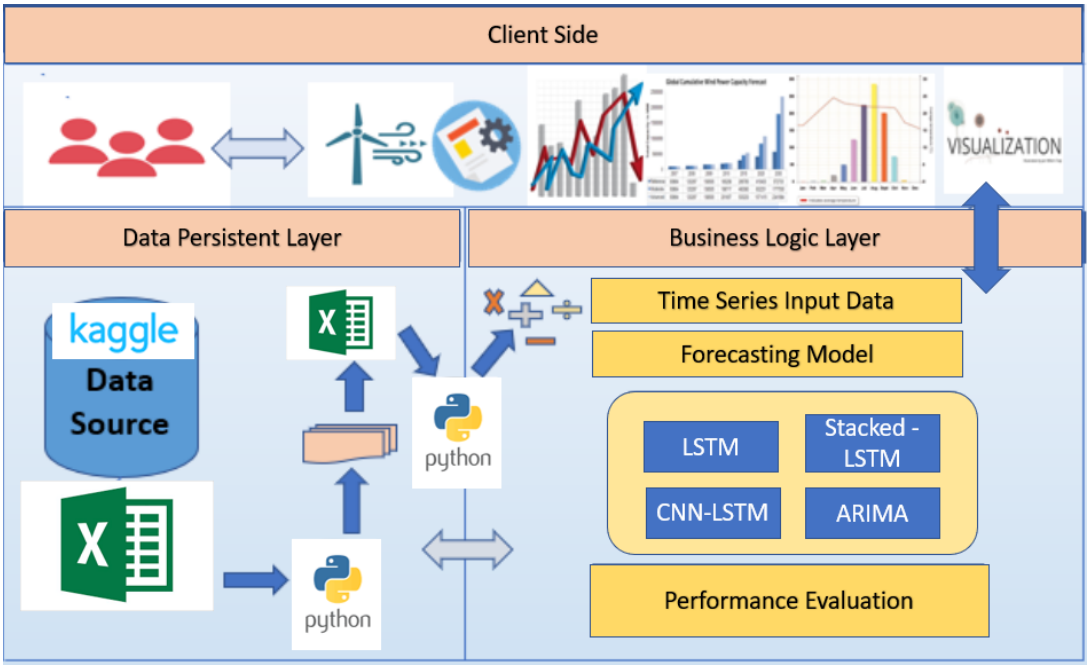


Figure 2: Project Architectural Design

3.5 Data Preprocessing

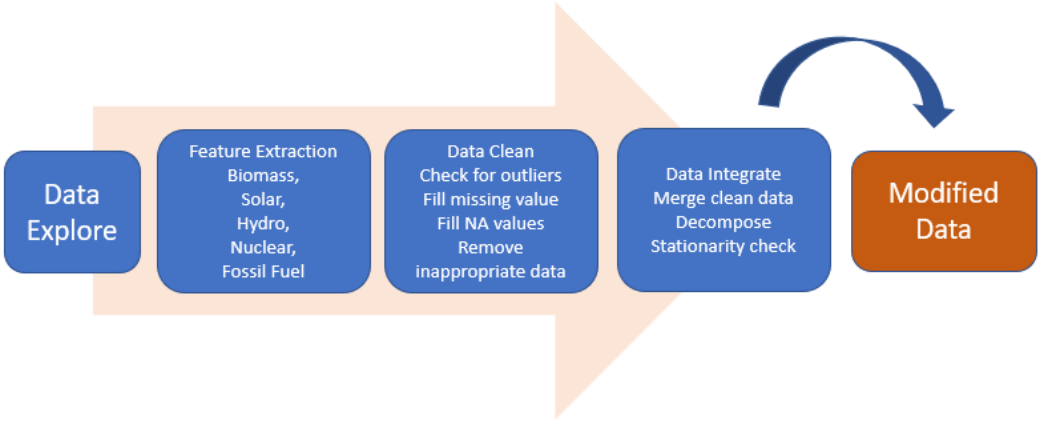


Figure 3: Data Preprocessing flow

A data frame using Pandas was created and loaded with an energy dataset. As only specific renewable sources like biomass, hydro water reservoir, nuclear, and solar are required for the analysis, other attributes were removed to avoid the overhead while running time series analysis. The time feature was set as an index considering the subsequent preprocessing and analysis. As by default, the time attribute holds string objects of time, so it was parsed to data time object using pandas function. Figure 3 represents the data flow diagram.

The data frame was further checked for missing data points. Since the missing value count in each consumption type was close to each other, it can be hypothesized that all

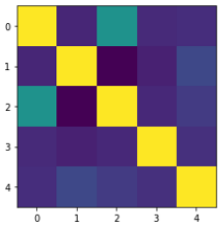
the missing values belong to the same record. To confirm this, every row with at least one missing value was displayed. It was observed that that hypothesis was correct. As there were missing values when compared to the total size of the data frame, imputing those values resulted in either of the below 3 options.

- Fill the empty cell with the average value
- Drop the entire row of data
- Find a better way of manipulating data points

```
In [8]: df.interpolate(method='time', limit_direction='forward', inplace=True, axis=0)
```

Figure 4: Data interpolation

The time series dataset has a property that data points will change with time. So, if the empty values are replaced with the mean value, then its seasonality is being destroyed. Dropping the entire record of data is also not possible as the time series data also has another property of consistent time value records. If records were dropped, it would erase some of the time interval entries which in turn disrupts the data oscillation on forecasting. One best way of data manipulation is to use interpolate, which creates a function for the available data values. New data values that are dependent on the past values are manipulated and then replaced with empty cells.



	generation biomass	generation hydro water reservoir	generation fossil hard coal	generation nuclear	generation solar
generation biomass	1.000000	-0.033307	0.433734	-0.021053	-0.004730
generation hydro water reservoir	-0.033307	1.000000	-0.157031	-0.049237	0.091661
generation fossil hard coal	0.433734	-0.157031	1.000000	-0.023150	0.045906
generation nuclear	-0.021053	-0.049237	-0.023150	1.000000	0.003911
generation solar	-0.004730	0.091661	0.045906	0.003911	1.000000

Figure 5: Correlation Matrix

Every attribute in the dataset was then examined with a correlation matrix for the relationship that they hold with other features. Logically, there may not be any correlation existing between any of the features, as the renewable source energies are independent and do not interrupt others. Figure 5 represents the correlation matrix, where the grid with the lightest color represents the strongest correlation. The correlation values were then analyzed and concluded that there is hardly any correlation between each feature in the dataset. The best could be between biomass and fossil hard coal which is still not too much, so it can be confirmed that all consumption types are independent of each other. The final cleaned and the reduced dataset was used for all the model algorithms.

3.6 Data Modelling

Tensorflow- Keras, and Statsmodels packages were used to implement the modeling of LSTM and ARIMA respectively. The preliminary step with the data modeling for time-series data will be seasonal decomposition. Time series includes components like overall

seasonality, observed trend, cyclic behavior, and noise in data. Decomposing the data helps to interpret each time component distinctly. All the four sources were decomposed and it was observed that there were some seasonality components that might exist. With these results, the dataset was examined further for the stationarity check.

```
biomass_season_period              = 30 * 24   # 30 days
nuclear_season_period              = 30 * 24   # 30 days
solar_season_period                = 1  * 24   # 1 day
fossilHardCoal_season_period       = 30 * 24   # 30 days
hydroWaterResorvoir_season_period  = 1  * 24   # 1 day
```

Figure 6: Seasonality analysis

The dataset contains around 4 years of data and while plotting the same, it seemed to have too many data points plotted in one screen. So, in the process of finding the seasonal period for each generation type took few zooming in of trends. For instance, biomass was analyzed to have 30 days of the seasonal period, solar consumption type had 1 day of seasonal period as more energy would be generated during day time and no energy during dawn. Figure 6 represents the summarized seasonal period.

LSTM, LSTM - CNN, Stacked- LSTM (Neural Network models), and ARIMA (Statistical model) models are used in forecasting the values in five different generation sources. With regards to LSTM, there is a three-way split of data performed, (i.e.) train set, validation set (used for hyperparameter), and test set. However, the dataset for the ARIMA model is divided into train and test as there is no hyperparameter involved here. A high-level API named Keras for the TensorFlow is used for the implementation of LSTM models. It gives user-friendly access to advanced deep learning frameworks. Both train and test sets were derived from the main dataset and MinMaxScaler from the library sklearn was used to scale the data.

3.7 Evaluation

As a first step in model performance evaluation, the results were plotted and visualized. The forecasted values were plotted along with the original or actual value in the dataset for a comparative study. The evaluation metrics used for this research are:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \tag{1}$$

$$\text{RMSErrors} = \sqrt{\frac{\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2}{n}} \tag{2}$$

$$\text{MAE} = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n} = \frac{\sum_{i=1}^{n}|e_i|}{n} \tag{3}$$

Figure 7: Evaluation Metrics

4 Design Specification

Preliminary data aggregation and data preprocessing were executed in Microsoft excel and Jupyter notebook in python. After a sequence of data cleaning and feature extraction activities, imputing and encodings were done. Once the dataset was finalized after the cleaning, different neural network models and ARIMA models were implemented in the same python Jupyter environment. Tensorflow framework was used with Keras in neural network model and statsmodel library for ARIMA to train the data points

4.1 LSTM

When it comes to the analysis of temporal data using recurring neural networks, then LSTM is one of the commonly used models. In this process, the input cell values are assigned with weights similar to each other. Weights of each cell are used to calculate the dot product. This in turn gets back an input vector. Figure 3 represents an LSTM cell that consists of the input gate, forget gate, and finally the output gate. The combined task of these gates is to compute the weights and connect values that are near to one or the other. With the use of Uniform Credit Assignment, vanishing gradients issues are being addressed.

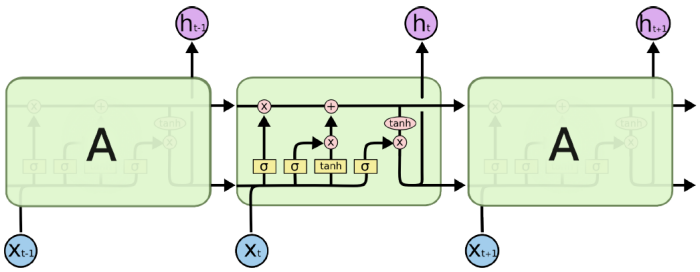


Figure 8: LSTM Cell

The above-described model is the uni-directional LSTM model, whereas bi-directional LSTM is also found to be greater in performance. The factor that distinguishes unidirectional from bi-directional is that bi-directional comprises of the future time periods while they are trained. The temporal sequences are processed in forwarding and backward direction every time before providing a single output. This further implies that the weights are calculated twice.

4.2 LSTM-CNN

In general Convolutional Neural Networks are being deployed for image (2 - dimensional) or video types of datasets. It is also efficient in extracting and gaining different metrics from any single variate time-series datasets. In this hybrid model, feature extraction is taken care of by the CNN from the input values. Followed by the output from CNN fed into the LSTM model as input. The complementary behaviors of CNN like extracting features that showcase spatial locality and LSTM for the temporal data on detecting features. However, the CNN model's accuracy might affect the LSTM model's performance. Hence, to eschew such discrepancies, CNN - LSTM parallel connectivity can be

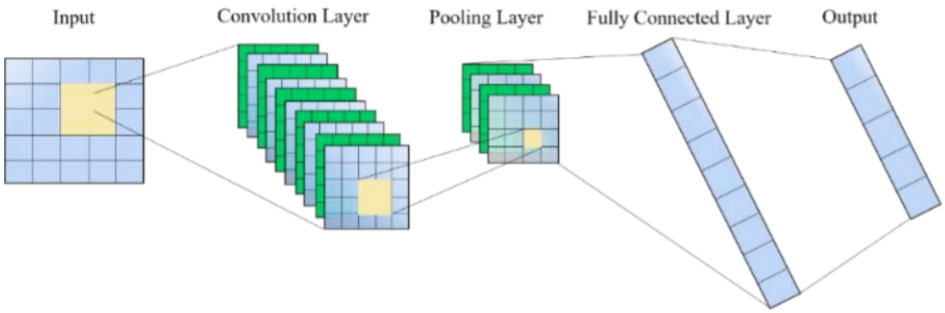


Figure 9: LSTM-CNN Architecture

Figure 10: LSTM-CNN flow

deployed where each neural network will have its own path without interfering with other networks. The output for the CNN path will be in the 2-dimensional array while its 1D array.

4.3 Stacked LSTM

A normal LSTM includes one hidden layer of LSTM and that leads to one output layer of feedforward. Whereas, stacked LSTM is the enhancement in regular LSTM, where it comprises multiple hidden layers of LSTM and every layer contains n number of memory cells. Stacking multiple LSTM layers will increase the depth of deep learning and provide a higher model accuracy rate. As the layers increase, the layer understands to combine and interpret the learnings that were made from the previous model and produce a new representations of abstracts. Increasing the depth to the layer is a form of optimization of representation.

Figure 11: Stacked LSTM Architecture

Performing LSTM once again beside the previous output as recurrent input will increase the model performance. This feed-forward architecture creates a hierarchical representation of features. To distinguish the presence of feedforward layer and stacked LSTM layer in between the input feature and LSTM layer is that a fully networked layer will not find the feedback from the previous frame and in turn, this cannot find few trends or patterns in the output layer. However, stacked LSTM recognizes complex patterns at every depth.

4.4 ARIMA

A standard model for time series analysis is ARIMA (Autoregressive Integrated Moving average) which is used highly in forecasting data points. This model is capable of predicting values based on historic data analysis and is mostly used for univariate as per (Noureen et al.; 2019). If any of the statistical metrics' live average or standard deviation does not have any temporal or spatial consistency, then the time series can be claimed to be weakly stationary. Arima could be delegated as:

ARIMA(p,d,q)

P- no of autoregressive terms

D- the nonseasonal difference

Q - moving average

A linear equation of regression model for any stationary time series is ARIMA. Where the lag terms of the actual values consist of predicted values.

5 Implementation

The implementation of all three neural network models and the ARIMA model used the same cleaned dataset with 6 attributes. This dataset had some empty values which were imputed using interpolate function. Unused attributes have been deleted from the dataset. Apart from scaling the data, no more processing of the dataset is required. The total records present are 34065 spanning from 2014 to 2018 (i.e) four years of data. The data is in hourly granularity.

All the neural network models implemented follow the same pattern of model definition, displaying the model architecture, training the created model, saving the model training, and evaluating of the loss and other parameters.

Table 1: LSTM Model Configuration

LSTM	
Parameter	**Value**
Input Layer	100
Hidden Layer	100
Dense Layer	relu
Batch Size	32
Epochs	120
Dropout	0.1
Loss	MSE

5.1 Implementation of LSTM

The preliminary data load was carried out by reading the cleaned CSV file into the environment. Considering the vulnerabilities of the LSTM model to be overfitting, a validation set hyperparameter was adopted. If hyperparameters were configured based on the test set, then it would likely be training the model with the test set. Hence, a three-way split is incorporated. Figure 1 represents the split ratio of the dataset. There are four years of data of which three years is for the train set 0.5 years is for validation

hyperparameter set and the last six months are for forecasting (i.e.) test set. The dataset was scaled using minmaxscaler function. This will consider the minimum value and the maximum value from the dataset and scale the data from 0 to 1 in this case. A window function that takes input as 2-dimensional arrays and returns a 3-dimensional output that maps 24 records of power generation to the 25th record (i.e.) for 24 hours of data the 25th-hour data is mapped as output. This window function is utilized in the dataset creation and scaler transformation steps. This window approach was adopted with reference to (Nair et al.; 2021).

The LSTM model architecture 13 was set up with one hidden layer of 100 neurons and an input layer of 100 neurons. The batch size is by default 32 and 0.2 was set up as the dropout. Relu function was used to optimize the model as illustrated in table 1

A graph is plotted to identify the loss function behavior between train and test validation sets. This study of loss leverages more understanding of where the loss curves meet each other. This will guide the number of epochs for the model implemented.

After performing the rescaling, the forecasted values were plotted against the actual values that belong to the same time period.

```
Total data is 4 years
Training is   3 years of data
Validation is 0.5 years of data
Testing is    0.5027397260273972 years of data
```

Figure 12: Three way split of dataset

```
=================================================================
the model for biomass generation:
Model: "lstm_biomass"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 24, 100)           40800
_________________________________________________________________
flatten_1 (Flatten)          (None, 2400)              0
_________________________________________________________________
dense_2 (Dense)              (None, 200)               480200
_________________________________________________________________
dropout_1 (Dropout)          (None, 200)               0
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 201
=================================================================
Total params: 521,201
Trainable params: 521,201
Non-trainable params: 0
_________________________________________________________________
```

Figure 13: Model Architecture of LSTM Biomass

5.2 Implementation of LSTM-CNN

With reference to the above implementation of LSTM 5.1 additional steps and workflow are added for LSTM-CNN implementation. On top of the regular LSTM model, the

convolutional layer is deployed and this extracts the low-level features. With regards to this study, hyperparameters are created namely the loss measurement with mean squared error. Other metrics like RMSE and MAPE are also assigned on different hyperparameters. Early stopping is used as it is aimed to have the loss to be the least for one model to be best. In an ideal case, the loss will be zero. But, it is never zero because in real scenarios noise in data is unavoidable.

The model keeps training the loss and the loss percentage gets reduced and at a point the loss becomes constant and at this point the model is stopped straining. This is done to avoid wastage of resources and the model is not expected to become any better.

Table 2: LSTM-CNN Model Configuration

CNN-LSTM	
Parameter	**Value**
Input Layer	100
Hidden Layer	100
Dense Layer	1
Conv1D filters	100
Activation	relu
Epochs	120
Loss	MSE

```
=================================================================
the model for biomass generation:
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 24, 100)           300
_________________________________________________________________
lstm (LSTM)                  (None, 24, 100)           80400
_________________________________________________________________
flatten (Flatten)            (None, 2400)              0
_________________________________________________________________
dense (Dense)                (None, 50)                120050
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 51
=================================================================
Total params: 200,801
Trainable params: 200,801
Non-trainable params: 0
_________________________________________________________________
```

Figure 14: Model Architecture of CNN-LSTM Biomass

5.3 Implementation of Stacked-LSTM

The implementation of stacked LSTM was carried out in the same steps as the LSTM and CNN LSTM executions 5.1. In addition to those steps, one LSTM layer is stacked on top of another LSTM layer. LSTM gives out multidimensional output when the return sequence hyperparameter is set to TRUE. The multidimensional output is reduced to one

dimensional using flatten function and then the data is passed on to a fully connected network that is the dense network.

It could be observed that one model is created for each generation type in every model implementation. This is because if one model was created for all the generation types, then the deep learning model will consider that the different generation sources are dependent on each other. But now as they are individually developed, it will be easy for the model to generalize the results.

Table 3: Stacked LSTM Model Configuration

Stacked LSTM	
Parameter	**Value**
Input Layer	100
LSTM Layer 1	100
LSTM Layer 1	50
Dense Layer	1
Activation	relu
Epochs	120
Dropout	0.1
Loss	MSE

```
=================================================================
the model for biomass generation:
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 24, 100)           40800
_________________________________________________________________
lstm_1 (LSTM)                (None, 24, 50)            30200
_________________________________________________________________
flatten (Flatten)            (None, 1200)              0
_________________________________________________________________
dense (Dense)                (None, 150)               180150
_________________________________________________________________
dropout (Dropout)            (None, 150)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 151
=================================================================
Total params: 251,301
Trainable params: 251,301
Non-trainable params: 0
```

Figure 15: Model Architecture of Stacked-LSTM Biomass

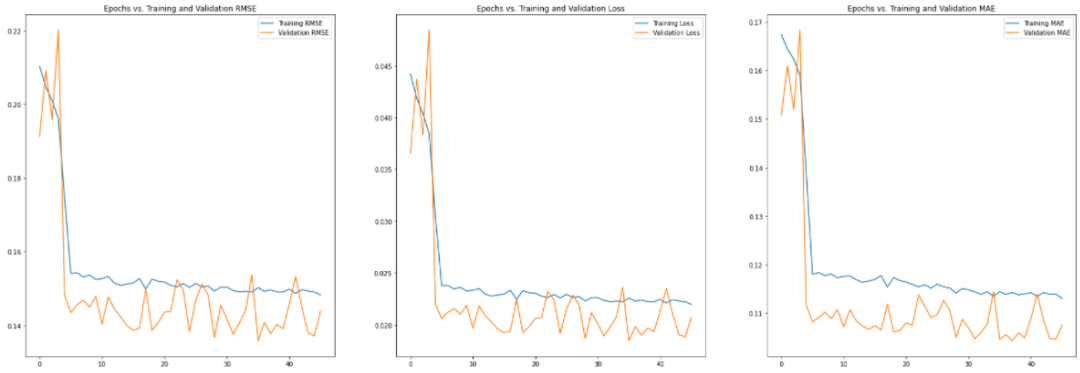


Figure 16: Validation Loss

5.4 Implementation of ARIMA

It is important for data to be stationary before being utilized for any statistical implementation models. Observations are seen as order as they are temporal in nature. Hence a form of consistency in the summary statistics is required and if not present, then it needs to be handled. The forecast is effective when the time series is stationary as they do not have any seasonality and trend effect. A unit root test named Augmented Dickey-Fuller test helped to understand the degree of trend influence on the time series data. A deep result observation was checked with the Autocorrelation Function that shows the relation with past data points and the Partial Autocorrelation Function, which shows relation with past mean residuals. Once these decompose and stationarity check was complete, it was confirmed that the data is stationary. If the data was non-stationary, then approaches like logging, deflating, or differencing would be used to bring stationarity in a time series.

The ARIMA model is trained on one year of data per generation type. The sliding window approach of training the model with 10 months of data and forecasting the next one hour is adopted.

6 Evaluation

6.1 LSTM

The LSTM model was trained on 120 epochs and achieved RMSE and MAE as indicated in 21. With regards to the prediction Vs actual data points, the model seemed to perform better for biomass, solar, and hydro water reservoir. But the other two sources were not seen with the generalized pattern as shown in Figure 17

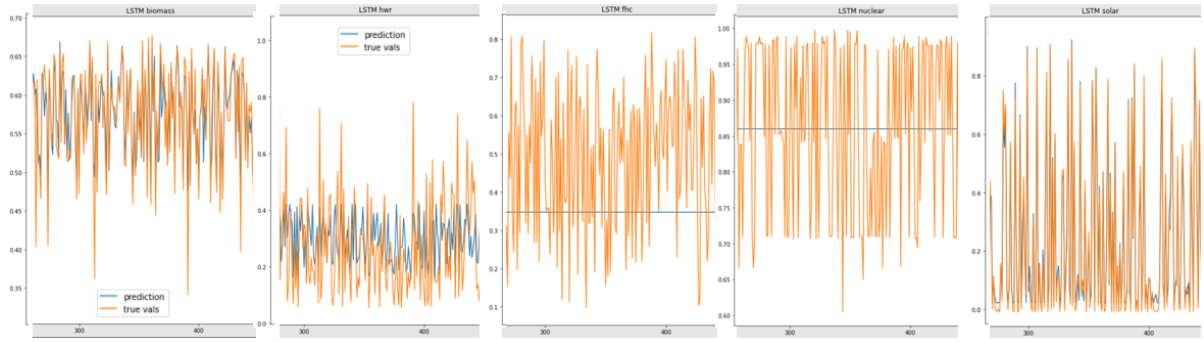


Figure 17: LSTM prediction Vs Actual

6.2 LSTM-CNN

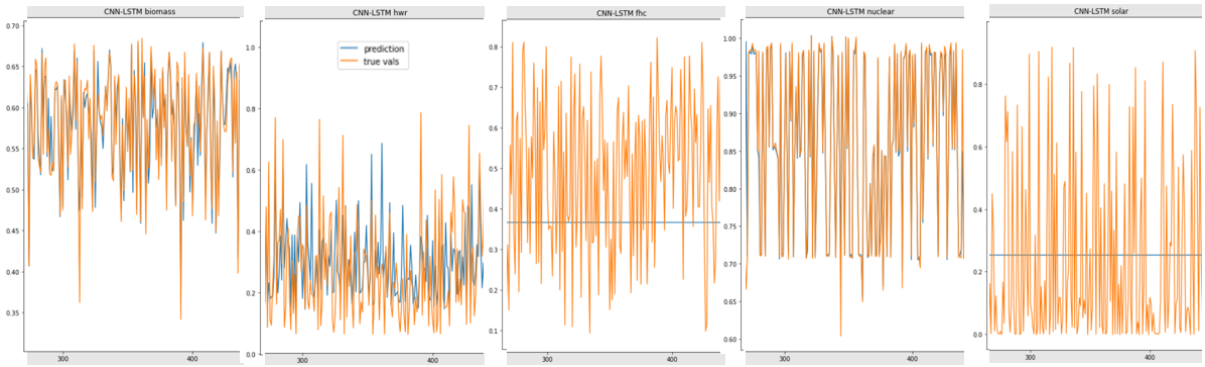


Figure 18: CNN-LSTM prediction Vs Actual

Similar to the LSTM, CNN-LSTM too was trained with 120 epochs and observed to have stopped with 35 epochs due to early stopping. From the prediction versus actual results 18 observations measure to have good generalization on biomass, hydro water reservoir, and nuclear.

6.3 Stacked-LSTM

Same as the other neural network models this stacked-LSTM was trained with 120 epochs. CNN-LSTM and Stacked LSTM proves to provide similar predictions.

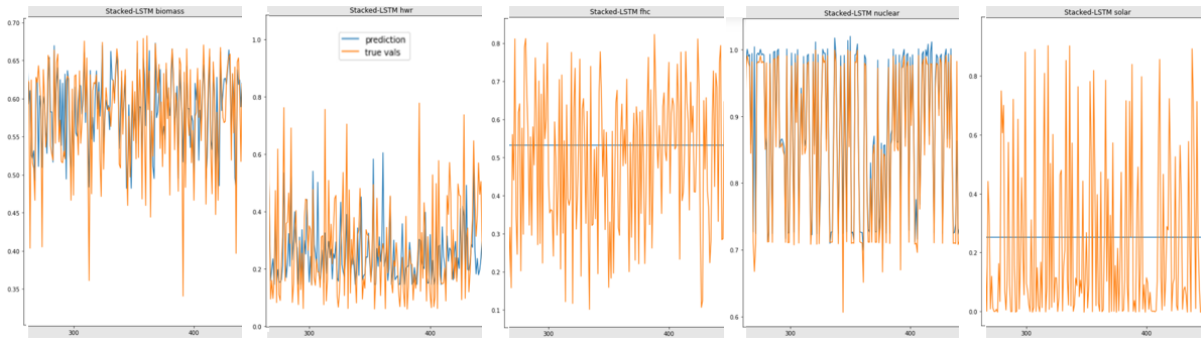


Figure 19: Stacked-LSTM prediction Vs Actual

6.4 ARIMA

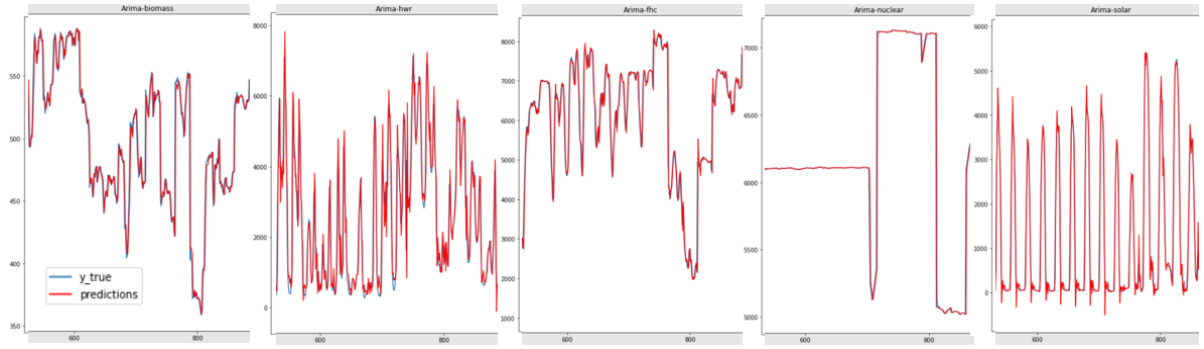


Figure 20: ARIMA prediction Vs Actual

ARIMA proves to be the best model as the predictions and actual values are significantly close and loss in ARIMA is very less.

Generation Type	MSE	RMSE	MAE
cnn_lstm_biomass	9.43E-04	0.030703	0.018231
cnn_lstm_hwr	1.77E-02	0.132997	0.104852
cnn_lstm_fhc	4.80E-02	0.219005	0.18588
cnn_lstm_nuclear	3.29E-04	0.01813	0.005645
cnn_lstm_solar	8.20E-02	0.286384	0.244694
lstm_biomass	1.42E-03	0.037628	0.024368
lstm_hwr	1.80E-02	0.134033	0.109201
lstm_fhc	5.36E-02	0.231579	0.195228
lstm_nuclear	1.24E-02	0.111575	0.09631
lstm_solar	1.42E-03	0.037709	0.029226
stacked_lstm_biomass	3.38E-03	0.058103	0.039278
stacked_lstm_hwr	1.73E-02	0.131459	0.099326
stacked_lstm_fhc	3.38E-02	0.183978	0.152968
stacked_lstm_nuclear	6.71E-04	0.02591	0.016577
stacked_lstm_solar	8.22E-02	0.286662	0.245897
ARIMA_biomass	3.02E+07	5494.798182	4472.078787
ARIMA_hwr	5.09E+07	7132.203994	5507.973425
ARIMA_fhc	5.39E+07	7341.298123	6157.616897
ARIMA_nuclear	2.66E+07	5157.850542	3877.560486
ARIMA_solar	2.56E+07	5063.289383	3705.724438

Figure 21: Error values of different generation type

6.5 Discussion

The utilization of neural network models for a time series dataset has many merits. The predictive abilities are added upon with the capacity of the temporal dimension. This helps the algorithm to analyze the current state of a given trend. For every point in the training of the dataset the output value is present in prior. Some extra overheads of computation that are associated with the LSTM do not imply that the training of the model takes more time. This adds on a degree of complex nature when compared to other simple deep learning algorithms. For instance, the neural network model requires the data to be rearranged on dimensionality. Forecasting energy generation from different renewable energy sources for 6 months was achieved with good results of acceptable error metrics.

The ARIMA model was recorded to have a better forecast rate when the time period of train data was reduced. Initially, 4 years of data were used to train the model. The training time was exponentially high, hence the input train data was reduced to 1 year for better model training.

With reference to the loss values from Figure 21, it can be interpreted that models with the lowest loss would have performed best. In the case of solar energy generation, the LSTM model outperforms the other neural networks. With regards to biomass, CNN LSTM records the lowest error in neural networks. Perhaps in the ARIMA model, almost all the model records have the lowest loss and showcase the highest forecasting rate.

7 Conclusion and Future Work

The aim of this research paper was to create statistical and neural network models for energy forecasting from different renewable energy sources. The models created and evaluated in this paper have achieved a good degree of success in answering the research question. All the four models LSTM, LSTM-CNN, stacked LSTM, and ARIMA have managed successfully to forecast long-term (6 months) data with a low loss rate. Each model performed better for different renewable sources. This was because of the different seasonality and noise behavior in the dataset. These forecasting models will help the private and governmental bodies to understand the energy capacity before even investing in any sector of power generation.

The inclusion of a weather dataset (that includes humidity, pressure, temperature, etc.) with the energy generation from renewable sources might help in deriving greater insights into the influence of climate on energy sources. This in turn will help in planning energy plants for further power production.

8 Acknowledgement

I sincerely thank Dr. Bharathi Chakravarthi for his guidance in the research work from stage 1 till the final report draft. The tips and tricks provided for the composition of the report were really helpful. This quality of research project was achievable because of the straightforward instructions and specification documents provided on different components in a project report available on moodle. Constructive feedbacks was very useful to make this research a better learning experience.

References

ArunKumar, K., Kalaga, D. V., Kumar, C. M. S., Kawaji, M. and Brenza, T. M. (2021). Forecasting of covid-19 using deep layer recurrent neural networks (rnns) with gated recurrent units (grus) and long short-term memory (lstm) cells, *Chaos, Solitons & Fractals* **146**: 110861.

Bantupalli, M. K. and Matam, S. K. (2017). Wind speed forecasting using empirical mode decomposition with ann and arima models, *2017 14th IEEE India Council International Conference (INDICON)*, pp. 1–6.

Bhatia, K., Mittal, R., Varanasi, J. and Tripathi, M. (2021). An ensemble approach for electricity price forecasting in markets with renewable energy resources, *Utilities Policy* **70**: 101185.

Cui, Z., Ke, R., Pu, Z. and Wang, Y. (2020). Stacked bidirectional and unidirectional lstm recurrent neural network for forecasting network-wide traffic state with missing values, *Transportation Research Part C: Emerging Technologies* **118**: 102674.

Gencer, K. and Başçiftçi, F. (2021). Time series forecast modeling of vulnerabilities in the android operating system using arima and deep learning methods, *Sustainable Computing: Informatics and Systems* **30**: 100515.

Jin, N., Yang, F., Mo, Y., Zeng, Y., Zhou, X., Yan, K. and Ma, X. (2022). Highly accurate energy consumption forecasting model based on parallel lstm neural networks, *Advanced Engineering Informatics* **51**: 101442.

Khan, W., Walker, S. and Zeiler, W. (2021). Improved solar photovoltaic energy generation forecast using deep learning-based ensemble stacking approach, *Energy* p. 122812.

Kumari, P. and Toshniwal, D. (2021). Long short term memory–convolutional neural network based deep hybrid approach for solar irradiance forecasting, *Applied Energy* **295**: 117061.

Li, B. and Haneklaus, N. (2021). The role of renewable energy, fossil fuel consumption, urbanization and economic growth on co2 emissions in china, *Energy Reports* **7**: 783–791.

Meenal, R., Selvakumar, A. I., Brighta, K., Joice, S. C. J. and Richerd, C. (2018). Solar radiation resource assessment using weka, *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, pp. 1038–1042.

Nair, A. S., Ranganathan, P., Finley, C. and Kaabouch, N. (2021). Short-term forecast analysis on wind power generation data, *2021 IEEE Kansas Power and Energy Conference (KPEC)*, pp. 1–6.

Natarajan, V. A. and Karatampati, P. (2019). Survey on renewable energy forecasting using different techniques, *2019 2nd International Conference on Power and Embedded Drive Control (ICPEDC)*, pp. 349–354.

Noureen, S., Atique, S., Roy, V. and Bayne, S. (2019). Analysis and application of seasonal arima model in energy demand forecasting: A case study of small scale agricultural load, *2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 521–524.

Paul, S., Dey, T., Saha, P., Dey, S. and Sen, R. (2021). Review on the development scenario of renewable energy in different country, *2021 Innovations in Energy Management and Renewable Resources(52042)*, pp. 1–2.

Ruhnau, O., Hennig, P. and Madlener, R. (2020). Economic implications of forecasting electricity generation from variable renewable energy sources, *Renewable Energy* **161**: 1318–1327.

Yahya, A. A. and Osman, A. (2019). Using data mining techniques to guide academic programs design and assessment, *Procedia Computer Science* **163**: 472–481.