



# **RENEWABLE POWER GENERATION FORECASTING**

being a dissertation submitted in partial fulfilment of the  
requirements for the degree of

Master of  
Artificial Intelligence & Data Science  
in the University of Hull

by  
Olumayowa Rominiyi (B.Sc.)  
August 2024

## Dedication

I would like to firstly acknowledge the Almighty God for His wisdom, faithfulness and the good health He has bestowed on me. I like to dedicate this research to my parents Mr & Mrs Rominiyi for their support morally & financially, and for believing in me, to my siblings for their encouragements. To Dr Zekun Guo for his constant availability and insightful guidance, to Mr Waheed Bello for his encouragement and guidance.

## Acknowledgements

We acknowledge the Viper High-Performance Computing facility of the University of Hull and its support team.

## Abstract

The use of fossil fuel is gradually being phased out because of the promises of the emerging renewable energy solutions. However, forecasting renewable power generation has posed significant challenges in its intermittency and unpredictability, which impacts their production and integration into the power grid. Precise forecasting is essential for power system operations. Methods developed over time to enhance renewable energy prediction accuracy include physical models, statistical approaches, artificial intelligence, and hybrid systems. This research presents the exploration of different approaches to renewable power forecasting by implementing a range of machine learning (ML) models, deep learning (DL) models and hybrid of deep learning models (HDL) performed on an open-source data obtained from a solar station in China for the year 2019 to 2020. This study analyses the different evaluation metrics obtained from the experimentation on the models. Among the models is a novel hybrid model of convolutional neural network (CNN), Bi-directional long short-term memory (BiLSTM), and transformer (TF) model - CNN-BiLSTM-TF which is an area not yet explored from the recently implemented Transformer network in hybrid model of CNN-LSTM-TF hybrid model. Results shows that CNN-LSTM-TF performed slightly better than BiLSTM counterpart having (MSE) of 25.19 and 29.43 respectively, (RMSE) of 5.02 and 5.43 respectively, R<sup>2</sup> Score of 0.88 and 0.86 respectively, (MAPE) of 85.94% and 117.61% all on the test set. The metrics became significantly better when the hybrid models were further hybridized using the best random forest (RF) model in this research, highlighting the robustness of random forest algorithm, and could pave the way for more accurate forecast in the future. This paper would be proving the efficacy of the recent combination of hybrid models to forecast renewable energy

Sections	Word Count
Introduction	265
Literature review	790
Methodology	1101
Result & Discussion	847
Conclusion	290
<b>Total</b>	<b>3293</b>

# Contents

Dedication.....	i
Acknowledgements .....	ii
Abstract .....	iii
List of Figures.....	vi
List of Figures (Appendix) .....	vii
List of Tables.....	viii
Chapter 1 Introduction.....	1
Chapter 2 Literature Review.....	2
2.1    Statistical Models .....	2
2.2    Machine Learning Models (ML) .....	2
2.3    Deep Learning Models .....	3
2.4    Hybrid models .....	3
Chapter 3 Methodology .....	4
3.1    Data Collection.....	5
3.2    Exploratory Data Analysis .....	5
3.3    Data Preprocessing .....	5
3.4    Machine Learning Models.....	7
3.4.1    Random Forest (RF) .....	7
3.4.2    XGBoost (XGB) .....	7
3.5    Deep Learning Models .....	8
3.5.1    Recurrent Neural Network (RNN).....	8
3.5.2    Long Short-Term Memory (LSTM) .....	8
3.5.3    Gated Recurrent Units (GRUs).....	8
3.5.4    Convolution Neural Network (CNN) .....	8
3.5.5    Transformers (TR) .....	8
3.6    Hybrid Deep Learning Models.....	8
3.7    Evaluation Metrics Selection.....	9
3.7.1    Mean Absolute Error (MAE) .....	9

3.7.2	Mean Squared Error (MSE) .....	9
3.7.3	Root Mean Squared Error (RMSE) .....	9
3.7.4	Coefficient of Determination( $R^2$ ).....	9
3.7.5	Mean Absolute Percentage Error (MAPE) .....	10
	Chapter 4 Results.....	11
4.1	Machine Learning Models.....	11
4.2	Deep Learning Models .....	13
4.3	Hybrid Models.....	14
	Chapter 5 Discussion .....	19
5.1	Machine Learning Models.....	19
5.2	Deep Learning Models .....	19
5.3	Hybrid Models.....	19
	Chapter 6 Conclusion.....	21
	Reference list / Bibliography .....	22
	Appendix 1 Model Results .....	I
	Appendix 2 Discussion .....	XXVIII

## List of Figures

Figure 1. Workflow & model selection flowchart.	4
Figure 2.(a) Average power by month of the year and (b) Heat map of Avg. Solar Power produced by hour & months.....	6
Figure 3. Correlation Heatmap for Feature Selection.	7
Figure 4. (a) Feature importance of Random Forest Model, (b) Scatter plot of predicted vs actual (Training set) and (c) Scatter plot of Predicted vs actual (Test set).....	11
Figure 5. (a) Feature importance of XGBoost Model, (b) Scatter plot of predicted vs actual (Training set), and (c) Scatter plot of Predicted vs actual (Test set).....	12
Figure 6. (a) RNN model architecture and (b) Training & Validation Loss plot .....	13
Figure 7(a) CNN-BiLSTM-TR model architecture, and (b) Training & Validation Loss plot.....	15
Figure 8 (a) CNN-BiLSTM-TR model actual vs predicted power on Test set, and (b) CNN-BiLSTM-TR-RF model actual vs predicted power on Test set.....	16
Figure 9 (a) Hybrid models comprehensive metrics on training set, and (b) Hybrid models comprehensive metrics on training set.....	17

## List of Figures (Appendix)

Figure A1. 1 Actual & predicted power on training and test set for RNN model.....	I
Figure A1. 2 Scatter plot of predicted vs actual on training and test set for RNN model .....	II
Figure A1. 3(a) LSTM model summary architecture (b) LSTM model training & validation loss plot .....	III
Figure A1. 4 Actual & predicted power on training and test set for LSTM model .....	IV
Figure A1. 5 Scatter plot of predicted vs actual power on training and test set for LSTM model .....	V
Figure A1. 6 (a) GRU model summary architecture (b) GRU model training & validation loss plot .....	VI
Figure A1. 7 Actual & predicted power on training and test set for GRU model.....	VII
Figure A1. 8 Scatter plot of predicted vs actual power son training and test set for GRU model .....	VIII
Figure A1. 9 (a) CNN model summary architecture (b) CNN model training & validation loss plot .....	IX
Figure A1. 10 Actual & predicted power on training and test set for CNN model.....	X
Figure A1. 11 Scatter plot of predicted vs actual power on training and test set for CNN model.....	XI
Figure A1. 12 (a) Transformers model summary architecture (b) model training & validation loss plot .....	XII
Figure A1. 13 Actual & predicted power on training & test set for transformers model .....	XIII
Figure A1. 14 Scatterplot of predicted vs actual power on training and test set for transformers model....	XIV
Figure A1. 15(a) CNN-LSTM model summary architecture (b) model training & validation loss plot .....	XV
Figure A1. 16 Actual & predicted power on training & test set for CNN-LSTM model .....	XVI
Figure A1. 17 Scatterplot of predicted vs actual power on training and test set for CNN-LSTM model .....	XVII
Figure A1. 18 Actual & predicted power on training & test set for CNN-LSTM-RF model.....	XVIII
Figure A1. 19 Scatterplot of predicted vs actual power on training and test set for CNN-LSTM-RF model ..	XIX
Figure A1. 20 (a) CNN-LSTM-TR model summary architecture (b) model training & validation loss plot .....	XXI
Figure A1. 21 Actual & predicted power on training & test set for CNN-LSTM-TR model.....	XXII
Figure A1. 22 Scatterplot of predicted vs actual power on training & test set for CNN-LSTM-TR model ...	XXIII
Figure A1. 23 Actual & predicted power on training & test set for CNN-LSTM-TR-RF model.....	XXIV
Figure A1. 24 Actual & predicted power on training & test set for CNN-BiLSTM-TR model.....	XXV
Figure A1. 25 Scatterplot of predicted vs actual power on training & test set for CNN-LSTM-TR-RF model .....	XXVI
Figure A1. 26 Actual & predicted power on training & test set for CNN-BiLSTM-TR-RF model.....	XXVII

## List of Tables

Table 1. Data columns, count and datatype.....	5
Table 2. Model evaluation metrics of Training & Test Set .....	18

# Chapter 1 Introduction

Renewable Energy Sources (RES) are crucial in transforming global energy. Wind and solar power are becoming vital for meeting worldwide energy demands. This shift to renewables is reshaping how we power our planet. Gielen et al. (2019) stated that “the share of renewable energy in the primary energy supply would rise from 14% in 2015 to 63% in 2050”. Renewables could meet 66% of global energy needs by 2050, significantly reducing greenhouse gases to keep global warming under 2°C. Energy availability is crucial for national economic development (Kumar et al., 2019).

RES variability challenges grid operators, making accurate forecasting crucial for system stability (Rhouma & Yahia Said, 2023). Precise load predictions enable better scheduling, maintenance, pricing, and contract assessment (Ruzic et al., 2003; Methaprayoon et al., 2003).

Renewable energy forecasting is typically classified by time horizon. Raza & Khosravi (2015) propose three categories: long-term (1-10 years), medium-term (1-12 months), and short-term (1 hour-1 week). Chen & Xu (2022) and Rahman & Hazim (1993) suggest four: very short-term (<30min), short-term (30min-6h), medium-term (6-24h), and long-term (1-7 days). Long-term forecasts aid expansion planning (Wang et al., 2022).

This research aims to predict short-term renewable power generation, particularly from solar stations. It focuses on Short-Term Load Forecasting (STLF), used for economic dispatch and unit mix studies, forecasting load conditions for upcoming hours (Kazemzadeh et al., 2020).

This paper's main contributions are: (1) Introducing two ML models, (2) Presenting various DL forecasting algorithms (ANN, RNN, LSTM, CNN) & Transformer and (3) Employing hybrid models to validate their effectiveness and enhance performance, as suggested by recent research.

## Chapter 2 Literature Review

Energy forecasting employs three approaches: physical models, traditional statistical methods ("black box"), and artificial intelligence (AI) models ("grey box") (Foley et al., 2012). Physical models, including Numerical Weather Prediction (NWP) and solar radiation models, rely on complex atmospheric data. For wind energy, they predict wind speed and direction; for solar, they account for factors like cloud cover affecting panel irradiance (Benti et al., 2023).

Traditional statistical methods like auto-regressive and moving average-based methods such as ARIMA and SARIMAX are popular for RES forecasting due to their seasonality and exogenous variable handling (Basmadjian et al., 2021). However, AI-based Machine Learning and Deep Learning algorithms have gained traction for their ability to learn complex data relationships (Benti et al., 2023). Generative adversarial networks (GANs), convolutional neural networks (CNN), long short-term memory (LSTM), and ensemble methods are widely used, with GANs excelling in capturing wind and solar energy fluctuations (Chen & Xu, 2022). ML & DL essentially map inputs to outputs by processing data and minimizing target value errors (Rhouma & Yahia Said, 2023).

### 2.1 Statistical Models

Statistical models are mostly applicable to small data problems of linear relationships and tend to be more interpretable (Dou et al., 2023). They include predictive approaches predominantly include pattern continuation, variable relationship studies, trend projection, and sequential data analysis (time series). Most research in the field of wind-generated power estimation employs sequential data analysis techniques (Das et al., 2018). The limitation of ARIMA is its inability to handle nonstationary series, which limits their utility in complex estimation tasks (Diagne et al., 2013). Researchers (Li et al., 2014) introduced an augmented analytical model for solar energy estimation (ARMAX). This model incorporates relevant environmental factors as exogenous inputs. Their findings showed that ARMAX could decrease the Root Mean Squared Error of two other methods by approximately 25%.

Basmadjian et al. (2021) found seasonal methods like SARIMAX outperform non-seasonal ARIMAX for most months. However, ARIMAX excels during periods of stable weather conditions.

### 2.2 Machine Learning Models (ML)

Traditional ML models excel in complex multivariate predictions and nonlinear relationships (Donadio et al., 2021; Rahman et al., 2021). ML outperforms statistical methods in complex power generation forecasting (Cocchi et al., 2018). While statistical methods offer interpretability, machine learning is often seen as "black box" (Kane et al., 2014; Alsaigh et al., 2023).

Renewable energy forecasting has progressed from early pattern recognition algorithms in SVMs (da Silva Fonseca Jr et al., 2012; Shi et al., 2012) and ANNs (Izgi et al., 2012; Dumitru et al., 2016; Son et al., 2018) to more sophisticated methods. Recent advancements include combined computational approaches, aggregate learning systems, and multi-layered frameworks, driven by the need for higher predictive accuracy (Dou et al., 2023).

ANNs have become prevalent in renewable energy forecasting. Nazir et al. (2020) note their attractiveness over other methods due to fewer limitations. ANNs are widely used in research for predictions, as they self-learn and adjust weightings to minimize errors between forecasts and actual results, enhancing their broad applicability.

## 2.3 Deep Learning Models

Deep learning models excel in handling complex problems and large datasets but requires significant computational resources. DL uses multi-layered ANNs to capture intricate data patterns (Wang et al., 2019). CNNs and RNNs are multi-layer ANNs; CNNs use convolutional, pooling, and fully connected layers to extract localized features (Akram et al., 2019). Neural network algorithms outperform ARIMA-based models (Kassa et al., 2016).

RNNs process sequential data, maintaining memory of previous elements (Hewamalage et al., 2021). They excel in NLP, speech recognition, and time series prediction. LSTM and GRUs are advanced RNN types that capture extended input patterns (Yang et al., 2020).

## 2.4 Hybrid models

Hybrid models combine strengths of various approaches, enhancing prediction accuracy. However, they face challenges like slow convergence, lack of control, and difficulty in selecting optimal models for combination (Dou et al., 2023). Composite techniques in renewable power output prediction frequently incorporate advanced pattern recognition algorithms and rapid learning frameworks. One prevalent approach utilizes a non-linear classification method based on kernel functions, which mitigates the risk of model overspecialization (Olatomiwa et al., 2015)

This research focuses on hybrid models, following Voyant et al.'s (2017) finding that SVR, SVM, ARIMA, and ANN excel in solar radiation forecasting. Huertas-Tato et al. (2020) showed SVM-based hybrids outperform single predictors. Gastón et al. (2010) combined K-NN and SVM, improving solar radiance prediction accuracy. ANN-SVM hybrids for PV forecasting yielded lower errors than individual ML models (Yang et al., 2014; Dong et al., 2015).

Recent hybrid models have improved PV power forecasting. Lim et al. (2022) combined LSTM and CNN, using CNN for weather classification and LSTM for power pattern learning. Salman et al. (2024) further enhanced this approach by adding a Transformer with attention, achieving 0.551% MAE and outperforming previous models.

## Chapter 3 Methodology

This section in the study discusses the approach taken to experiment on the forecasting of renewable energy as shown in Figure 1.

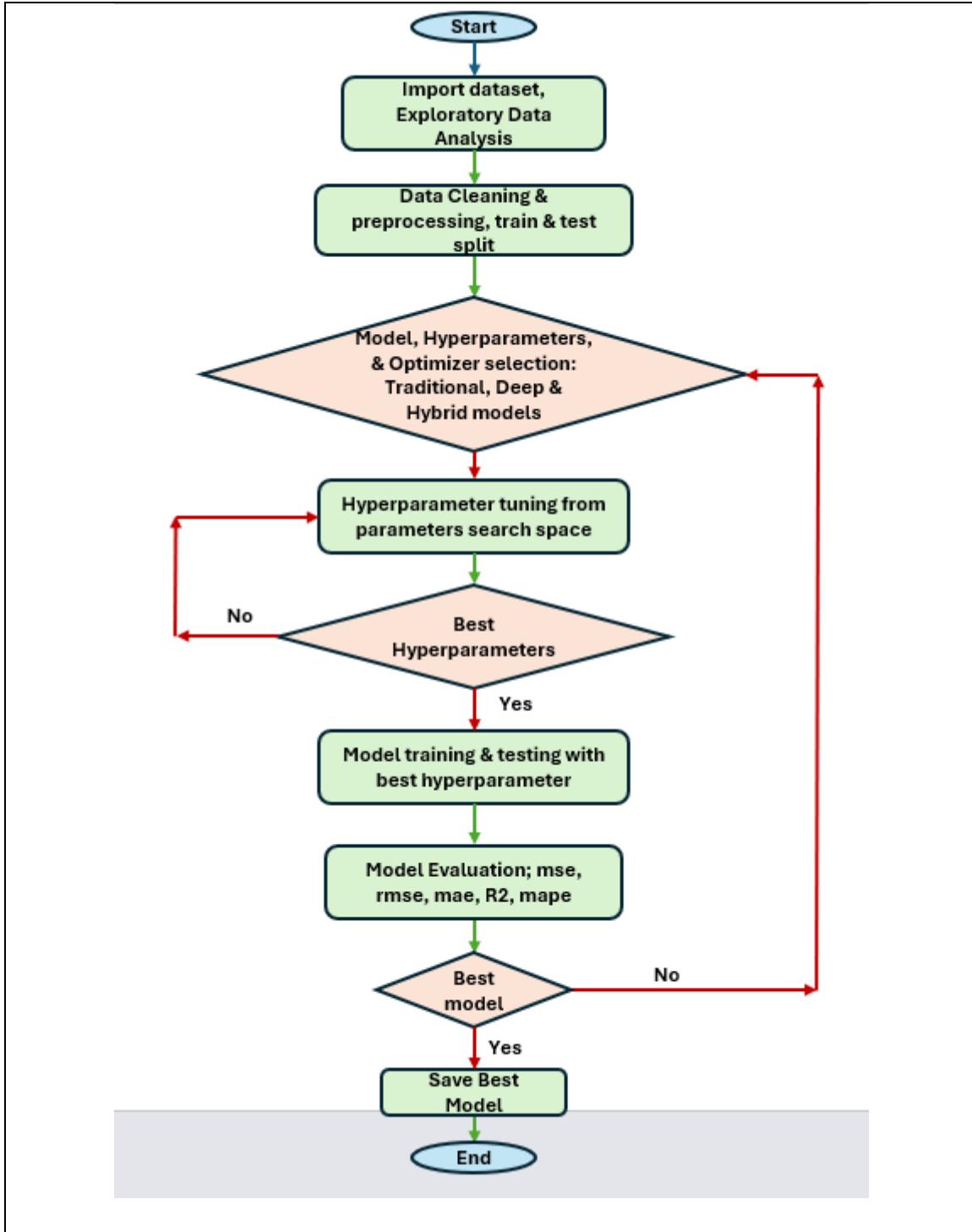


Figure 1. Workflow & model selection flowchart.

### 3.1 Data Collection

The dataset chosen for this study is an open-source dataset sourced from an on-site renewable energy station, comprising data collected from eight (8) solar stations and six (6) wind farms located in China and spanned over a duration of 2 years (2019-2020) and recorded at 15-minutes intervals. The 1<sup>st</sup> solar station with maximum output power of 50MW was selected for this study. The solar station data set comprises of 7 columns (Total Solar Irradiance(W/m<sup>2</sup>), Direct Normal Irradiance(W/m<sup>2</sup>), Global Horizontal Irradiance(W/m<sup>2</sup>), Air Temperature(°C), Atmosphere(hpa), Relative Humidity (%), Power (MW)).

### 3.2 Exploratory Data Analysis

Table 1. Data columns, count and datatype

S/N	Columns	Non-Null Count	Datatype
1	Total Solar Irradiance (W/m <sup>2</sup> )	70,176	Int64
2	Direct Normal Irradiance (W/m <sup>2</sup> )	70,176	Int64
3	Global Horizontal Irradiance (W/m <sup>2</sup> )	70,176	Int64
4	Air Temperature (°C)	70,176	Float64
5	Atmosphere (hpa)	70,176	Float64
6	Relative Humidity (%)	70,176	Float64
7	Power (MW)	70,176	Float64

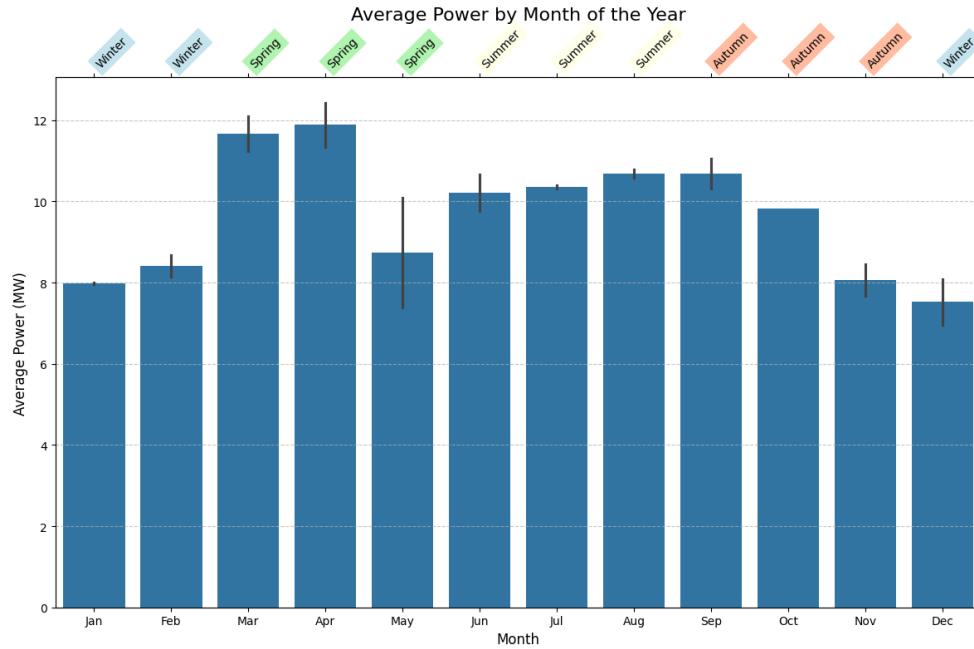
The data comprise of 70,176 non-null values as shown in Table 1 with 60 outliers. Figure 2 indicates that Solar power generation closely follows daily and seasonal solar irradiance patterns. Daily production starts at dawn, peaks midday, and ends at dusk. Seasonally, spring and summer months, particularly March and April, show the highest output with average peaks around 12 kW. Summer afternoons often exceed 30 kW. Winter months see significantly reduced generation, with daily peaks of 15-20 kW and monthly averages of 7-8 kW. This stark seasonal contrast reflects variations in daylight duration and sun angle throughout the year.

### 3.3 Data Preprocessing

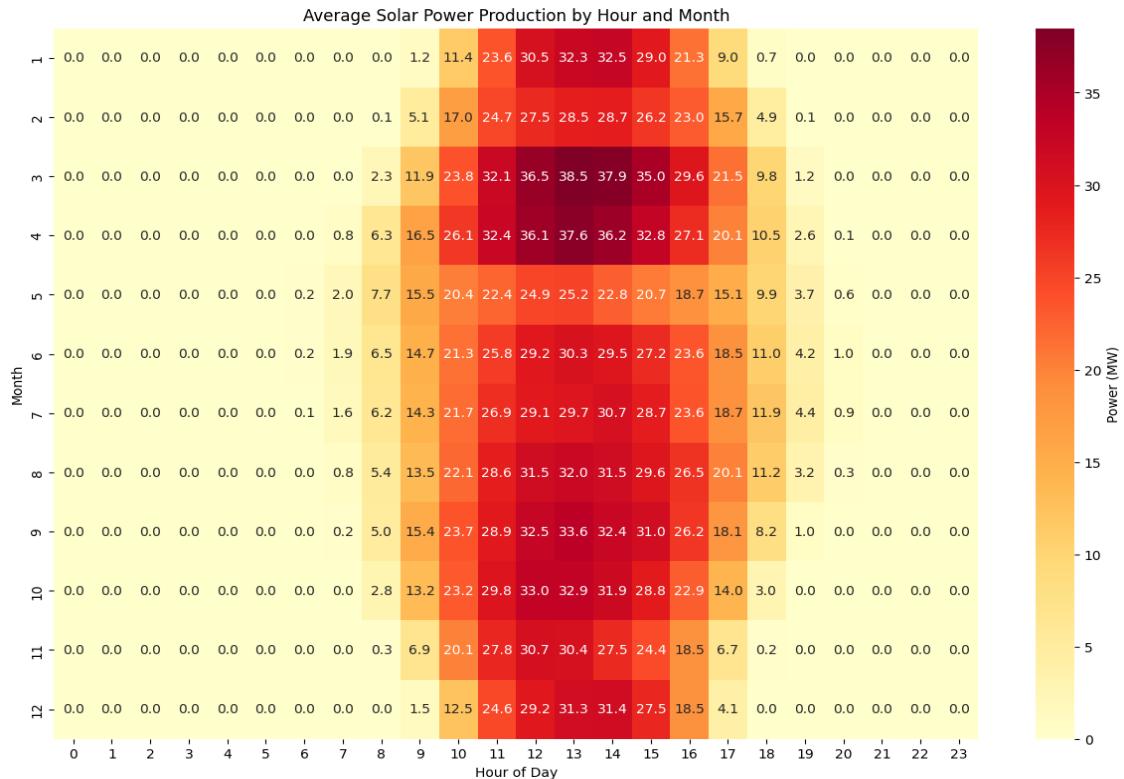
The following data science approaches were implemented before the forecasting experimentation and this including outliers (-99) values in the selected solar station dataset were interpolated with forward filled and the justification for this process is because the interval of 15min was a short duration for excess change to occur, so there is a higher chance each data point had similar characteristics as its neighbors.

The subset of 30,000 rows from the 70,176 rows was selected, in consideration to high computational expense and split into 80% (24,000) train set and 20% (6,000) test set. A correlation matrix shown in Figure 3, was used to obtain the highest correlated features (Total Solar Irradiance, Direct Normal Irradiance, Global Horizontal Irradiance & Air Temperature(°C)) for the feature engineering to predict

the output (Power). The train and test set were scaled using Sklearns standard scaler to ensure that feature sharing the same scale contribute equally to the model and for easier computations.



(a) Average Power by Month of the year



(b) Heat map of Average Solar power production by hour & months

Figure 2.(a) Average power by month of the year and (b) Heat map of Avg. Solar Power produced by hour & months.

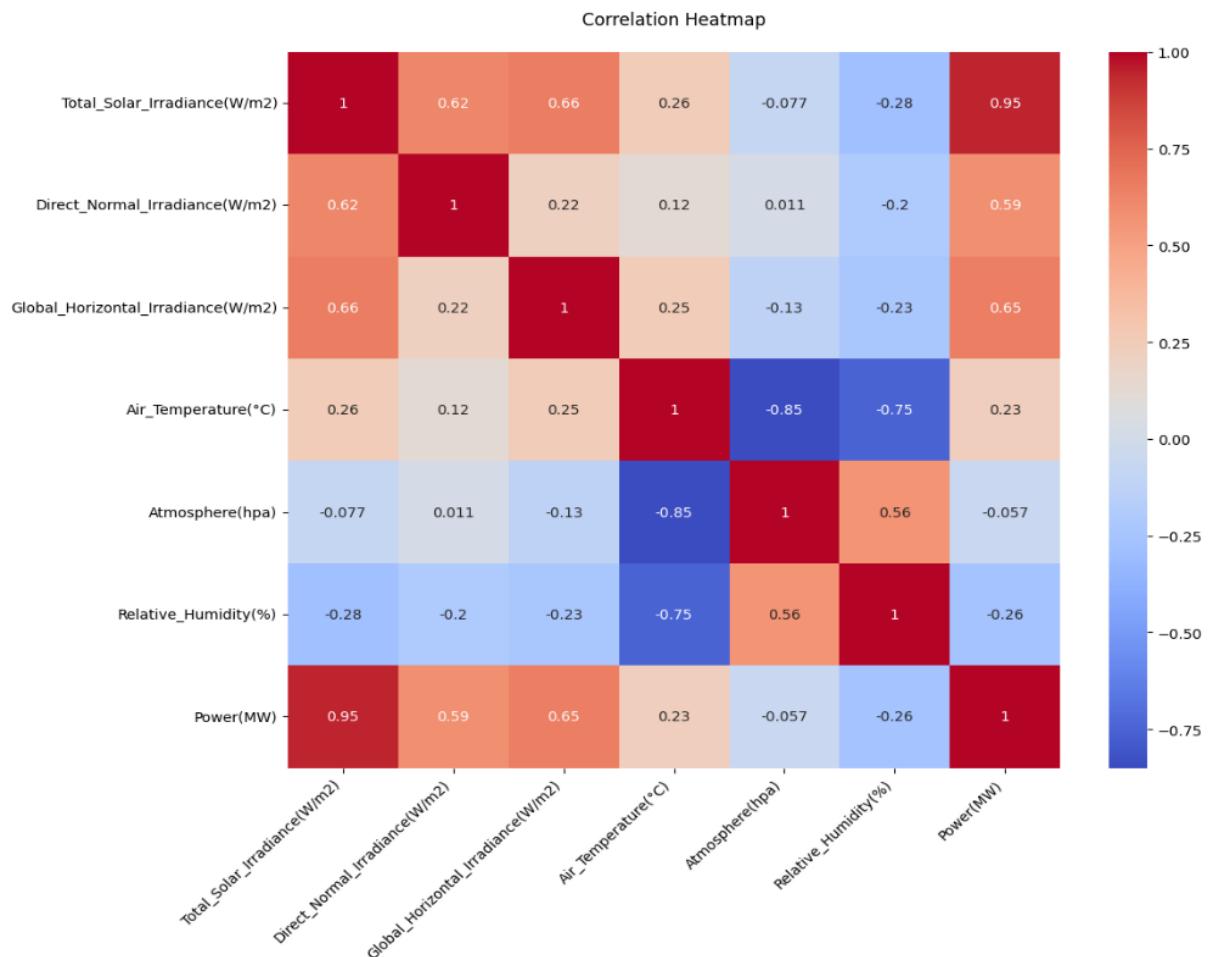


Figure 3. Correlation Heatmap for Feature Selection.

## 3.4 Machine Learning Models

### 3.4.1 Random Forest (RF)

RandomizedSearchCV was used to determine the best parameters for the Random Forest Model from the following hyperparameters. n\_estimators: [100, 200, 300], max\_depth: [None, 10, 20, 30], min\_samples\_split: [2, 5, 10], min\_samples\_leaf: [1, 2, 4], max\_features: [1, 2, 3, sqrt, log2].

### 3.4.2 XGBoost (XGB)

RandomizedSearchCV was used to determine the best parameters for the XGBoost Regression Model from the following hyperparameters. n\_estimators: [100, 200, 300], max\_depth: [3, 4, 5, 6], learning\_rate: [0.01, 0.1, 0.2], subsample: [0.8, 0.9, 1.0], colsample\_bytree: [0.8, 0.9, 1.0]

## 3.5 Deep Learning Models

The deep learning models implemented are RNN, LSTM, GRU, CNN, TRANSFORMERS. The common hyperparameter for all DL models are, dense\_layers: [1, 2], dense\_units: [16, 32], dropout\_rate: [0.1, 0.2, 0.3], learning\_rate: [1e-4, 1e-3, 1e-2], optimizer: [adam, nadam, rmsprop]. All modes were executed over 50 epochs.

### 3.5.1 Recurrent Neural Network (RNN)

The optimal hyperparameters for the RNN was determined from the following search space. Rnn\_units: [32, 64], num\_rnn\_layers: [1, 2].

### 3.5.2 Long Short-Term Memory (LSTM)

The optimal hyperparameters for the LSTM was determined from the following search space. Lstm\_units: [32, 64], num\_lstm\_layers: [1, 2].

### 3.5.3 Gated Recurrent Units (GRUs)

The optimal hyperparameters for the GRU was determined from the following search space. gru\_units: [32, 64], num\_gru\_layers: [1, 2].

### 3.5.4 Convolution Neural Network (CNN)

The optimal hyperparameters for the GRU was determined from the following search space. conv\_filters: [32, 64], kernel\_size: [3,5], num\_conv\_layers: [1, 2].

### 3.5.5 Transformers (TR)

The optimal hyperparameters for the Transformers was determined from the following search space. num\_transformer\_blocks: [1, 2], num\_heads: [2, 4], key\_dim: [16, 32], ff\_dim: [32, 64],

## 3.6 Hybrid Deep Learning Models

Hybrid Models combines the strength of different architectures to harness complementary features and enhance overall performance. CNN-LSTM-TF hybrid model in Salman et al. study (2024) gave the best result in which the model emphasizes the transformer component using self-attention to focus on the most important temporal contexts while capturing both local and global dependencies. This study aims to implement and improve on the accuracy of the hybrid model of CNN-BiLSTM-TF. All models were executed over 50 epochs. The hybrid models were further hybridized by stacking method, where the prediction of each model was passed as an input into the best random forest model, this was an interesting exploration because the hybrid of the following was achieved; CNN-LSTM, CNN-LSTM-RF, CNN-LSTM-TR, CNN-LSTM-TR-RF, CNN-BiLSTM-TR, CNN-BiLSTM-TR-RF.

## 3.7 Evaluation Metrics Selection

These are quantitative measures used to determine the effectiveness of a predictive model or algorithm and to enable the comparison of multiple models by giving objective criteria that can be used to assess the efficacy of a model. This study employs the following metrics.

### 3.7.1 Mean Absolute Error (MAE)

The average of absolute prediction errors. Its less sensitive to outliers than MSE and provides a straightforward measure of error magnitude (Salman et al. 2024), as shown in equation 3.1

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i| \quad 3.1$$

### 3.7.2 Mean Squared Error (MSE)

The average of squared prediction errors, emphasizing larger mistakes. Its sensitive to outliers and useful for penalizing significant errors (Salman et al. 2024), as shown in equation 3.2

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2 \quad 3.2$$

### 3.7.3 Root Mean Squared Error (RMSE)

The square root of MSE, providing error measurement in the same units as the target variable. It balances error emphasis and interpretability (Salman et al. 2024), as shown in equation 3.3

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2} \quad 3.3$$

### 3.7.4 Coefficient of Determination( $R^2$ )

A measure of the proportion of variance in the dependent variable explained by the independent variables, ranging from 0 to 1. It indicates how well the model fits the data with 1 being a perfect fit (M. Elsaraiti & A. Merabet, 2022), as shown in equation 3.4

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad 3.4$$

### 3.7.5 Mean Absolute Percentage Error (MAPE)

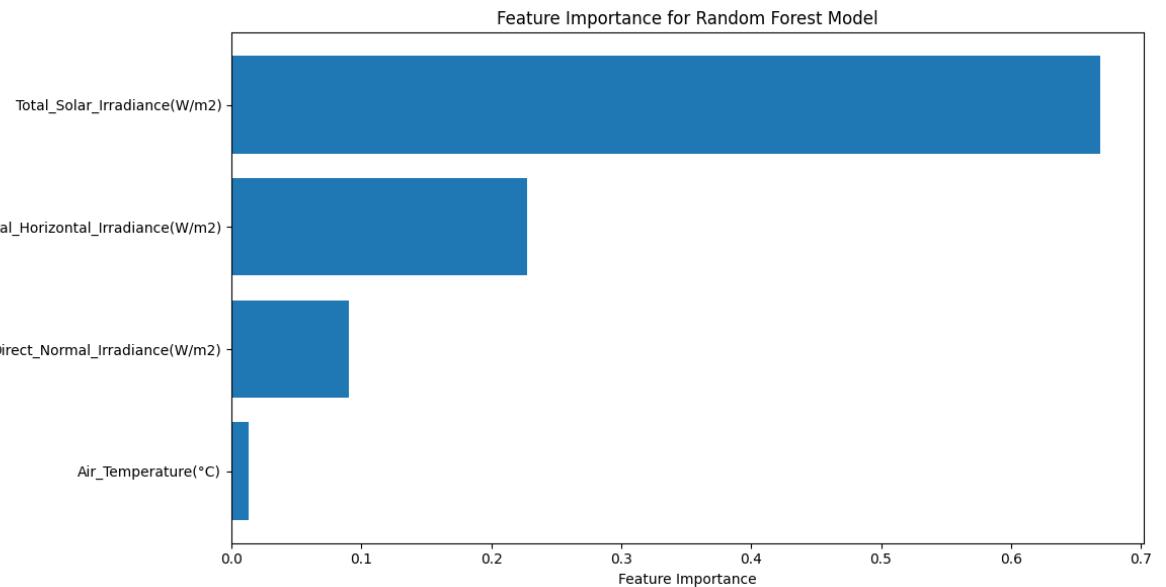
The average of absolute percentage errors, expressing accuracy as a percentage. It's useful for comparing errors across different scales but can be problematic with values near zero. (M. Elsaraiti & A. Merabet, 2022), as shown in equation 3.5

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{\bar{y}_i} \right| \times 100\% \quad 3.5$$

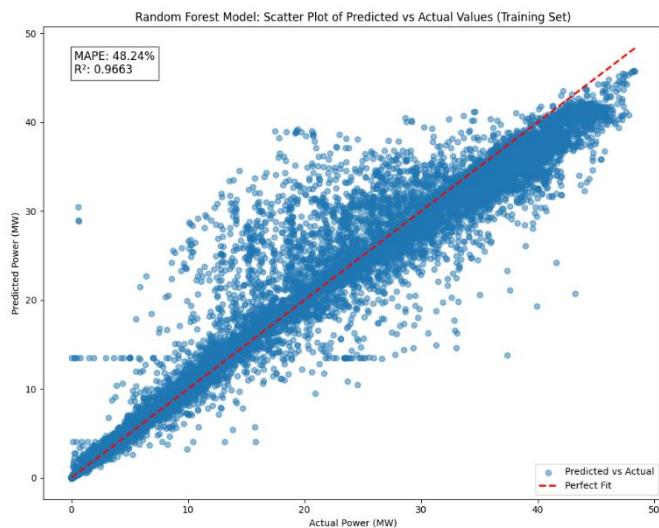
Where  $y_i$  is the actual value of the dependent variable for the  $i$ th observation,  $\hat{y}_i$  is the predicted value of the depended variable for the  $i$ th observation,  $\bar{y}$  is the mean of the actual values of the dependent variable and N is the number of observations.

# Chapter 4 Results

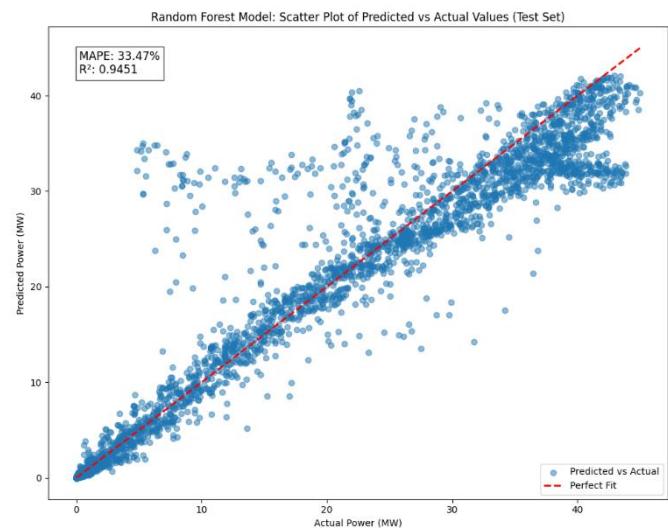
## 4.1 Machine Learning Models



(a) Feature importance plot of RF model

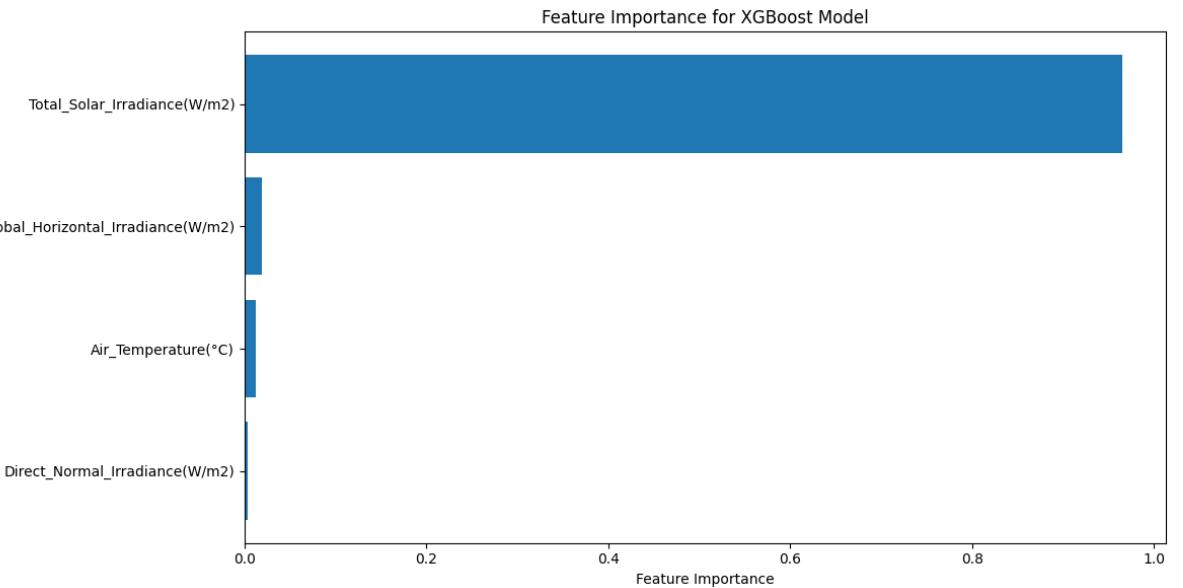


(b) Scatter plot of Predicted Vs Actual (Training)

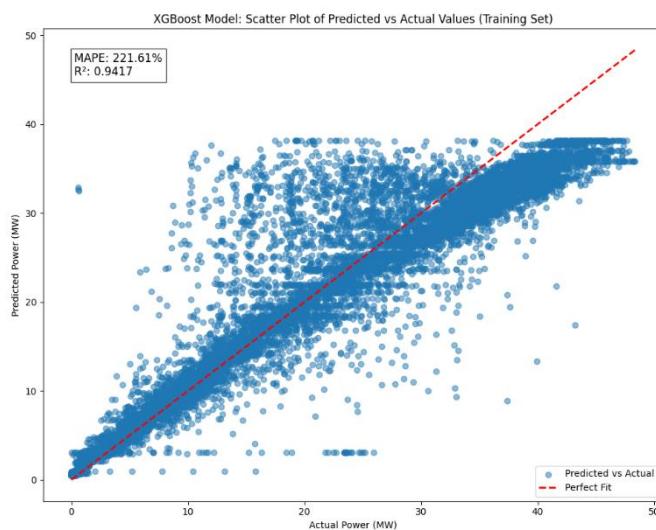


(c) Scatter plot of Predicted Vs Actual (Test)

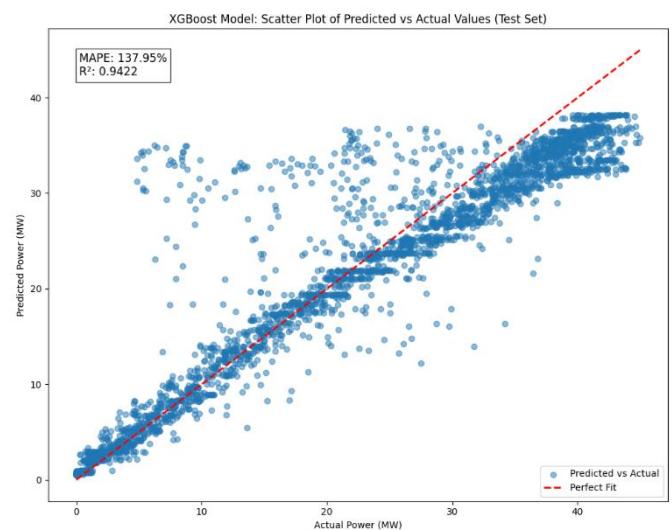
Figure 4. (a) Feature importance of Random Forest Model, (b) Scatter plot of predicted vs actual (Training set) and (c) Scatter plot of Predicted vs actual (Test set)



(a) Feature importance plot of XGBoost model



(b) Scatter plot of Predicted Vs Actual (Training)



(c) Scatter plot of Predicted Vs Actual (Test)

Figure 5. (a) Feature importance of XGBoost Model, (b) Scatter plot of predicted vs actual (Training set), and (c) Scatter plot of Predicted vs actual (Test set)

Random Forest model considers more features across the 4 selected features than the XGBoost model that is more concentrated on the total solar irradiance, as shown in Figure 4a and Figure 5a. Figure 4b & Figure 4b shows that random forest predicts better as there are more datapoints closer to the line of perfect fit, than xgboost model as shown in Figure 5a & Figure 5b

## 4.2 Deep Learning Models

Model: "sequential\_1"

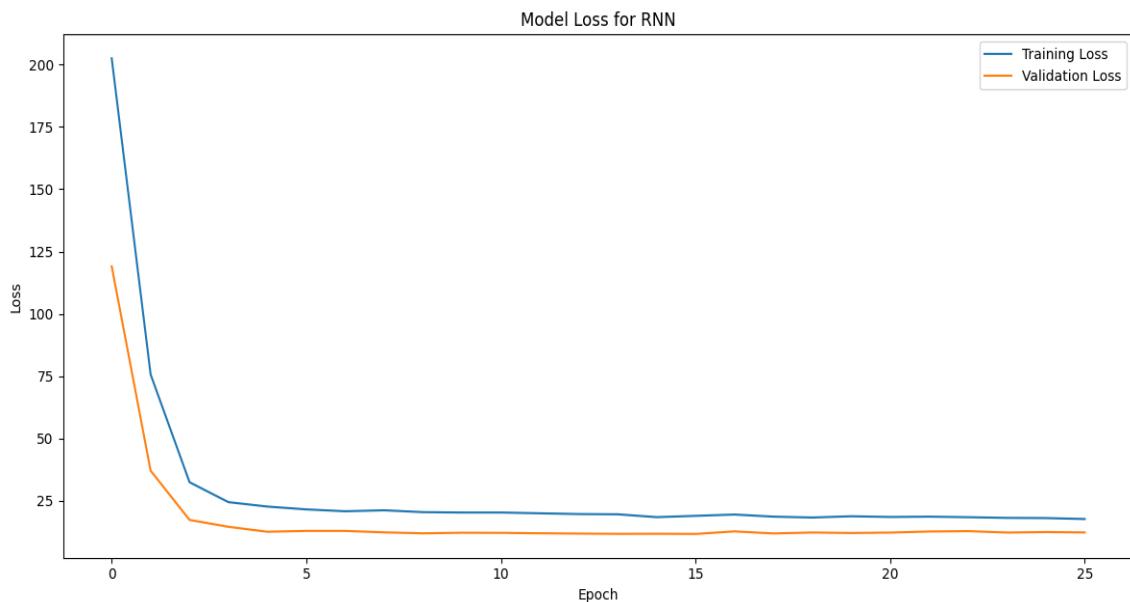
Layer (type)	Output Shape	Param #
simple_rnn_2 (SimpleRNN)	(None, 96, 64)	4,416
dropout_3 (Dropout)	(None, 96, 64)	0
simple_rnn_3 (SimpleRNN)	(None, 32)	3,104
dropout_4 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 16)	528
dropout_5 (Dropout)	(None, 16)	0
dense_3 (Dense)	(None, 1)	17

Total params: 8,065 (31.50 KB)

Trainable params: 8,065 (31.50 KB)

Non-trainable params: 0 (0.00 B)

(a) RNN Model Architecture



(b) RNN Model Training & Validation Loss

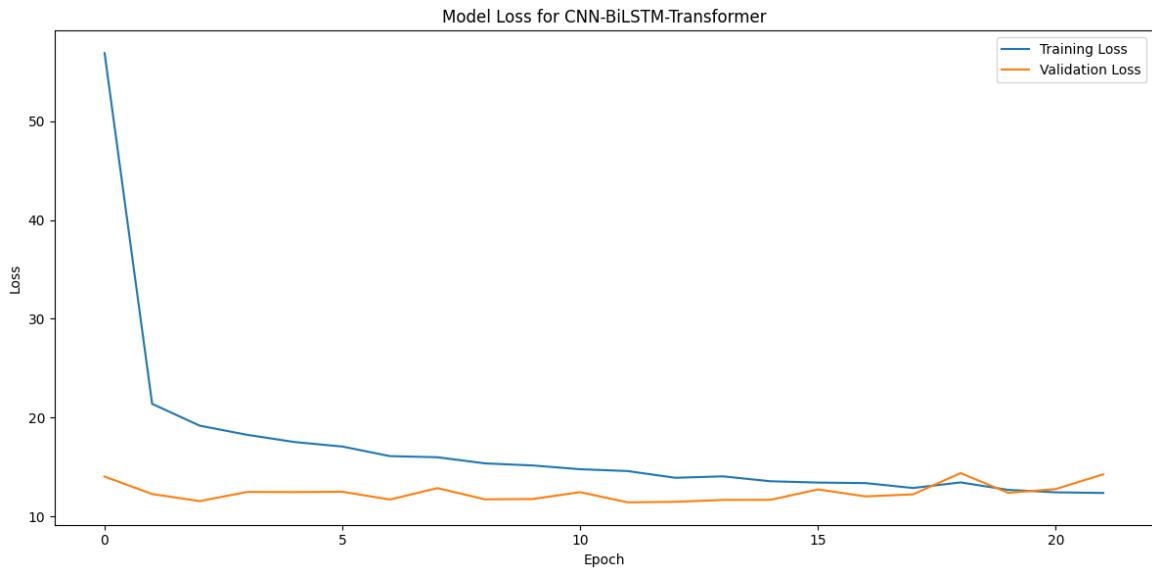
Figure 6. (a) RNN model architecture and (b) Training & Validation Loss plot

### 4.3 Hybrid Models

Model: "functional_1"			
Layer (type)	Output Shape	Param #	Connected to
input_layer_1 (InputLayer)	(None, 96, 4)	0	-
conv1d_2 (Conv1D)	(None, 96, 16)	208	input_layer_1[0][0]
max_pooling1d_2 (MaxPooling1D)	(None, 48, 16)	0	conv1d_2[0][0]
conv1d_3 (Conv1D)	(None, 48, 32)	1,568	max_pooling1d_2[0][0]
max_pooling1d_3 (MaxPooling1D)	(None, 24, 32)	0	conv1d_3[0][0]
bidirectional_1 (Bidirectional)	(None, 24, 128)	49,664	max_pooling1d_3[0][0]
dropout_9 (Dropout)	(None, 24, 128)	0	bidirectional_1[0][0]
multi_head_attention_2 (MultiHeadAttention)	(None, 24, 128)	33,088	dropout_9[0][0], dropout_9[0][0]
dropout_11 (Dropout)	(None, 24, 128)	0	multi_head_attention_2[0][0]
add_4 (Add)	(None, 24, 128)	0	dropout_9[0][0], dropout_11[0][0]
layer_normalization_4 (LayerNormalization)	(None, 24, 128)	256	add_4[0][0]
dense_7 (Dense)	(None, 24, 32)	4,128	layer_normalization_4[0][0]
dense_8 (Dense)	(None, 24, 128)	4,224	dense_7[0][0]
dropout_12 (Dropout)	(None, 24, 128)	0	dense_8[0][0]
add_5 (Add)	(None, 24, 128)	0	layer_normalization_4[0][0], dropout_12[0][0]
layer_normalization_5 (LayerNormalization)	(None, 24, 128)	256	add_5[0][0]
multi_head_attention_3 (MultiHeadAttention)	(None, 24, 128)	33,088	layer_normalization_5[0][0], layer_normalization_5[0][0]
dropout_14 (Dropout)	(None, 24, 128)	0	multi_head_attention_3[0][0]
add_6 (Add)	(None, 24, 128)	0	layer_normalization_5[0][0], dropout_14[0][0]
layer_normalization_6 (LayerNormalization)	(None, 24, 128)	256	add_6[0][0]
dense_9 (Dense)	(None, 24, 32)	4,128	layer_normalization_6[0][0]
dense_10 (Dense)	(None, 24, 128)	4,224	dense_9[0][0]
dropout_15 (Dropout)	(None, 24, 128)	0	dense_10[0][0]
add_7 (Add)	(None, 24, 128)	0	layer_normalization_6[0][0], dropout_15[0][0]
layer_normalization_7 (LayerNormalization)	(None, 24, 128)	256	add_7[0][0]
global_average_pooling1d_ (GlobalAveragePooling1D)	(None, 128)	0	layer_normalization_7[0][0]
dense_11 (Dense)	(None, 64)	8,256	global_average_pooling1d_[0][0]
dropout_16 (Dropout)	(None, 64)	0	dense_11[0][0]
dense_12 (Dense)	(None, 64)	4,160	dropout_16[0][0]
dropout_17 (Dropout)	(None, 64)	0	dense_12[0][0]
dense_13 (Dense)	(None, 1)	65	dropout_17[0][0]

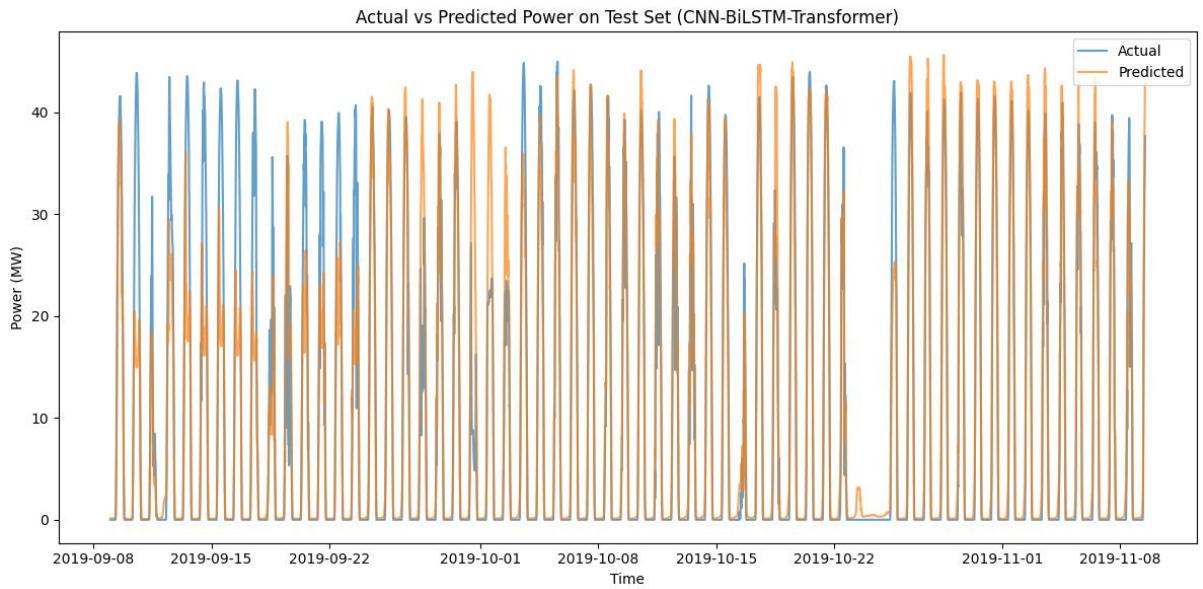
Total params: 147,825 (577.44 KB)  
Trainable params: 147,825 (577.44 KB)  
Non-trainable params: 0 (0.00 B)

(a) CNN-BiLSTM-TR Model Architecture

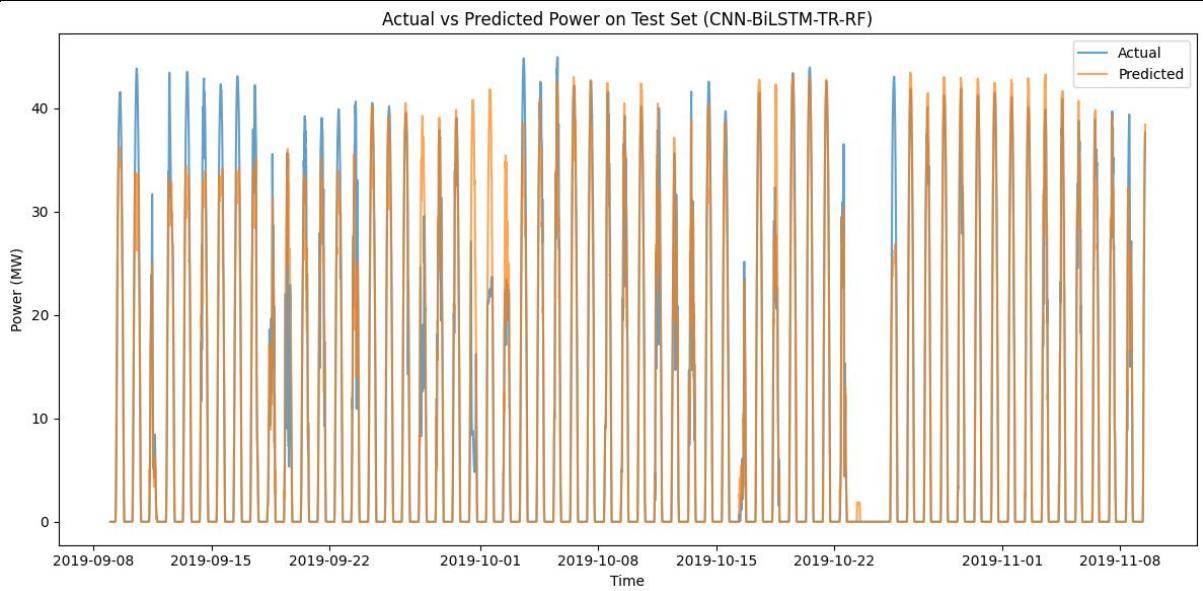


(b) CNN-BiLSTM-TR Training & Validation Loss

Figure 7(a) CNN-BiLSTM-TR model architecture, and (b) Training & Validation Loss plot

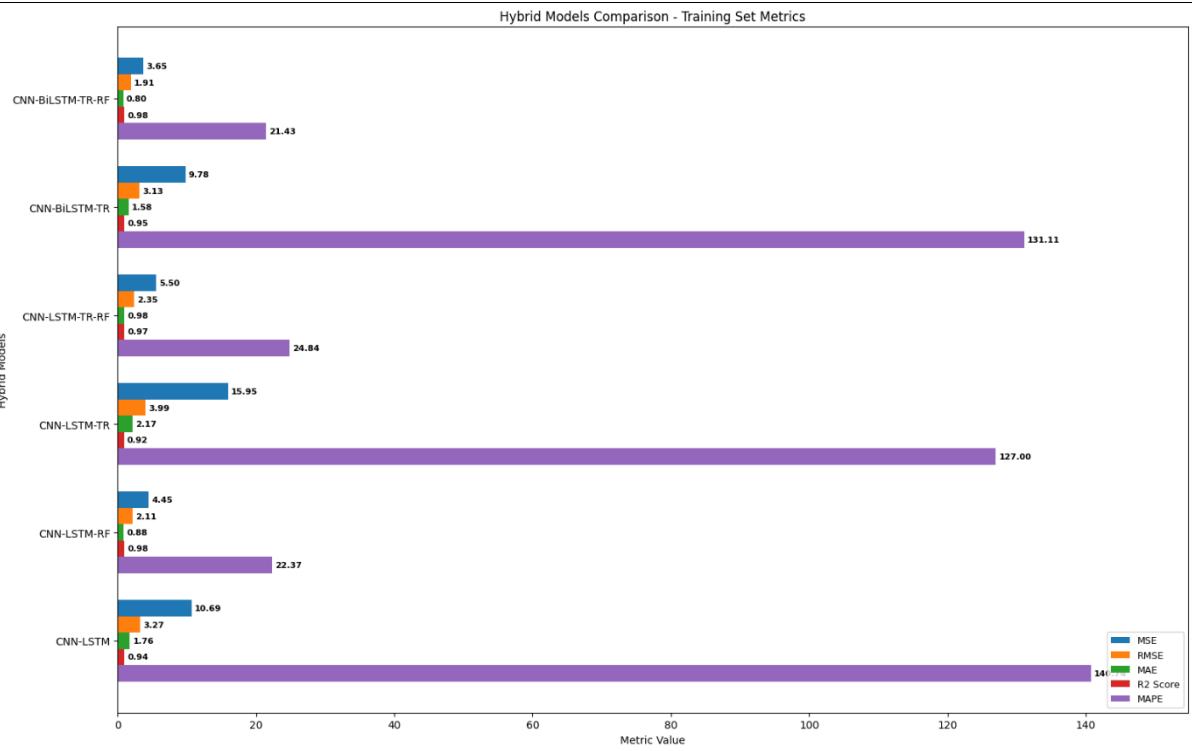


(a) Plot of actual vs predicted of CNN-BiLSTM-TR (without further hybridization with Random Forest)

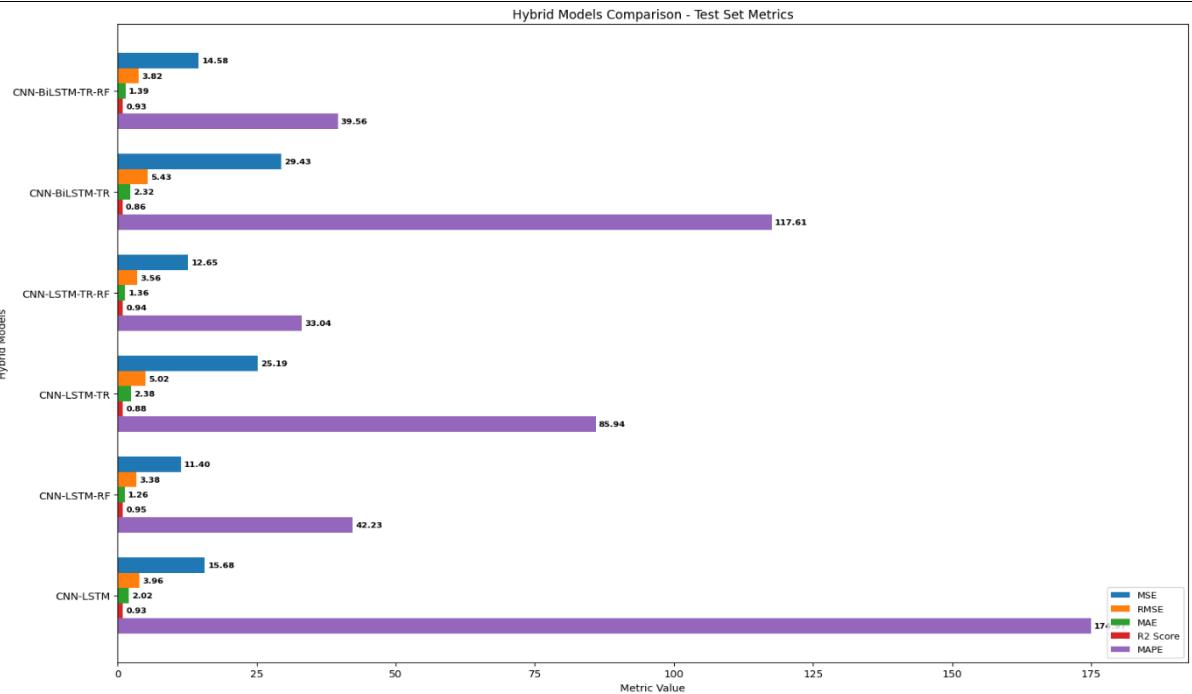


(b) Plot of actual vs predicted of CNN-BiLSTM-TR-RF (further hybridization with Random Forest)

Figure 8 (a) CNN-BiLSTM-TR model actual vs predicted power on Test set, and (b) CNN-BiLSTM-TR-RF model actual vs predicted power on Test set



(a) Hybrid Models Comprehensive Metrics Comparison on Training set



(b) Hybrid Models Comprehensive Metrics Comparison on Test set

Figure 9 (a) Hybrid models comprehensive metrics on training set, and (b) Hybrid models comprehensive metrics on training set

Table 2. Model evaluation metrics of Training & Test Set (Best results in each section are bold)

Models	MSE (Train)	RMSE (Train)	MAE (Train)	R2 Score (Train)	MAPE (Train) %	MSE (Test)	RMSE (Test)	MAE (Test)	R <sup>2</sup> Score (Train)	MAPE (Test) %
<b>Machine Learning Models</b>										
Random Forest	<b>6.45</b>	<b>2.54</b>	<b>1.07</b>	<b>0.97</b>	<b>48.24</b>	<b>11.45</b>	<b>3.38</b>	<b>1.34</b>	<b>0.95</b>	<b>33.47</b>
XGBoost	11.16	3.34	1.80	0.94	221.61	12.04	3.47	1.76	0.94	137.95
<b>Deep Learning Models</b>										
RNN	11.28	3.36	1.61	0.94	<b>94.89</b>	<b>11.97</b>	<b>3.46</b>	<b>1.48</b>	<b>0.94</b>	<b>73.56</b>
LSTM	11.17	3.34	1.70	0.94	138.80	12.01	3.47	1.60	0.94	110.53
GRU	<b>9.81</b>	<b>3.13</b>	<b>1.51</b>	<b>0.95</b>	122.90	13.72	3.70	1.71	0.93	92.46
CNN (CONV1D)	17.57	4.19	2.42	0.91	231.70	27.70	5.26	2.99	0.87	282.18
Transformer	180.45	13.43	11.08	0.06	2591.21	211.21	14.53	11.82	0.00	1464.34
<b>Hybrid Models</b>										
CNN-LSTM	10.69	3.27	1.76	0.94	140.74	15.68	3.96	2.02	0.93	174.97
CNN-LSTM-RF	4.45	2.11	0.88	<b>0.98</b>	22.37	<b>11.40</b>	<b>3.38</b>	<b>1.26</b>	<b>0.95</b>	42.23
CNN-LSTM-Transformer	15.95	3.99	2.17	0.92	127.00	25.19	5.02	2.38	0.88	85.94
CNN-LSTM-TR-RF	5.50	2.35	0.98	0.97	24.84	12.65	3.56	1.36	0.94	<b>33.04</b>
CNN-BiLSTM-Transformer	9.78	3.13	1.58	0.95	131.11	29.43	5.43	2.32	0.86	117.61
CNN-BiLSTM-TR-RF	<b>3.65</b>	<b>1.91</b>	<b>0.80</b>	<b>0.98</b>	<b>21.43</b>	14.58	3.82	1.39	0.93	39.56

## Chapter 5 Discussion

### 5.1 Machine Learning Models

Figure 4(a) demonstrates that the Random Forest model is more robust and considers more features than the XGBoost model in Figure 5(a), Table 2 indicated that Random Forest model has a lesser MAPE of 48.24% and 33.47% for the training and test set respectively compared to the XGBoost which are 221.61% and 137.95% respectively which shows signs of overfitting. The scatter plot shows predictions clustered tightly around the perfect fit line across most of the power range, with some increased spread for higher power values (>35 MW) on training set, the scatter plot shows a similar pattern to the training set but with more spread in predictions, particularly noticeable in the mid to high power ranges (20-40 MW) on the test set as seen in Figure 4b &Figure 4c for random forest model, while for XGBoost model, the scatter plot shows a good distribution of predictions around the perfect fit line, with some overestimation tendency for mid-range values (15-30 MW) and underestimation for higher values (>35 MW) on the training set and similar pattern to the training set, with some increased spread in predictions, particularly for higher power values (>30 MW). The  $R^2$  score for Random Forest is 0.97 training and 0.95 test set compared to a lesser 0.95 training, while the  $R^2$  for Xgboost is 0.94 across training and test set.

### 5.2 Deep Learning Models

Although the GRU has performed better on the training set given MSE 9.81, RMSE 3.13, MAE 1.51,  $R^2$  0.95 as shown in Table 2, The best model in the DL section is RNN with the optimal hyperparameters of 2 RNN layers with 64 units, a dropout rate of 0.1, 1 dense layer with 16 units as indicated in the model architecture, and the training & validation loss for 25 epochs as shown in Figure 6. Adam optimizer performed better, giving the results of having MSE 11.28, RMSE 3.36, MAE 1.61,  $R^2$  0.94 and MAPE 94.89 for training set, and MSE 11.97, RMSE 3.46, MAE 1.48,  $R^2$  0.94 and MAPE 73.56.

The Transformers model significantly underperforms with the optimal hyperparameters of 1 transformer block with 4 heads, 32 key dimensions, 32 forward feed dimensions, 1 dense layer with 32 units each and Adam optimizer obtaining MSE 180.45, RMSE 13.43, MAE 11.08, very low  $R^2$  of 0.06 and an outrageous MAPE 2591.21 % on training and MSE 211.21, RMSE 14.53, MAE 11.82,  $R^2$  of 0.00 and MAPE 1464.34% as shown in Table 2, the model summary architecture, and training & validation loss for 10 epochs is shown in Figure A1. 12. These metrics signifies that transformer model on its own cannot forecast the power output, hence the hybridization was further implemented. The underperformance of transformers model is further displayed in the poor prediction plot as shown in Figure A1. 13, and Figure A1. 14 shows the datapoint are horizontal and not on the line of perfect fit.

### 5.3 Hybrid Models

The hybrid model of CNN, LSTM, Transformer and Random Forest with their architecture and possible performance throughout the experimentation process has proven quite challenging and interesting.

The optimal hyperparameters for the proposed novel model of CNN-BiLSTM-Transformer are 2 CNN layer with 16 & 32 filter respectively, kernel size 3 each, 1 BiLSTM layer with 64 units, 2 Transformer block with 2 attention head having 32 key dimensions and 32 feed forward dimension units each, 2 dense layers with 64 units, Adam optimizer and a learning rate of 0.00039. Figure 7 shows the summary architecture, and the training & validation loss plot over 20 epochs. Table 2 shows the proposed model had the following metrics across the training set with MSE 9.78, RMSE 3.13, MAE 1.58,  $R^2$  0.95, MAPE 131.11 for training and MSE 29.43, RMSE 5.43, MAE 2.32,  $R^2$  0.86, MAPE 117.61 for the test set.

The plot of the metrics for the hybrid models in Figure 9 shows that for every further hybridization of the models with the best random forest model, the prediction was better on the training set and generalizes better on the test set. These significant differences can be observed in the actual vs predicted power plot of the proposed model before and after the inclusion of random forest as shown in Figure 8. CNN-BiLSTM-TR-

RF had the best metrics on the training set with MSE 3.65, RMSE 1.91, MAE 0.80,  $R^2$  0.98, MAPE 21.43, and MSE 14.58, RMSE 3.82, MAE 1.39,  $R^2$  0.93, MAPE 39.56 for test set which are significantly lesser than the aforementioned metrics of CNN-BiLSTM-TR as shown in Table 2.

Further discussion about the other models can be found in the Appendix section.

## Chapter 6 Conclusion

Renewable energy is crucial for future power generation and the earth's ecological progress. However, the intermittent and unpredictable nature of these natural power source prevents its effective incorporation into the power grid. Different Machine learning and Deep learning models have been implemented to curtail this problem. Hybrid models, however difficult to build, have performed and indicated better results because it harnesses the different strengths of the individual models.

Random forest model has been shown to perform better in handling non-linear relationships and complex data and provides accurate predictions regardless of outliers than XGBoost model. Hybrid models, particularly those incorporating Random Forest (RF), tend to perform best overall. The CNN-BiLSTM-TR-RF model shows the lowest error metrics on the training set. Random forest hybrid models consistently achieve the lowest MAPE, suggesting it handles percentage errors well across different scales of the target variable. The novel approach of further hybridization with Random Forest or any other model that could be used as Meta model is an aspect of forecasting whose advantages could be further harnessed and implemented in the future as it shows promising results.

This study found hybrid models superior to individual and deep learning models. Adam optimizer generally outperformed RMSprop and Nadam in most deep learning and hybrid models, with exceptions in GRU, CNN, and CNN-LSTM which utilized the RMSprop optimizer. The Nadam optimizer was not used in any of the optimal hyperparameters selections.

The challenge of computational power could be tackled to allow for more dataset; hence the full dataset of the solar station and the wind farm dataset could not be explored in this research, however the methodology of the study involves saving the models which could be used for similarly pre-processed dataset in the future.

## Reference list / Bibliography

Akram, M. W., Li, G., Jin, Y., Chen, X., Zhu, C., Zhao, X., Khaliq, A., Faheem, M. & Ahmad, A. (2019) CNN based automatic detection of photovoltaic cell defects in electroluminescence images. *Energy*, 189 116319.

Alsaigh, R., Mehmood, R. & Katib, I. (2023) AI explainability and governance in smart energy systems: A review. *Frontiers in Energy Research*, 11 1071291.

Basmadjian, R., Shaafieyoun, A. & Julka, S. (2021) Day-ahead forecasting of the percentage of renewables based on time-series statistical methods. *Energies*, 14 (21), .

Benti, N. E., Chaka, M. D. & Semie, A. G. (2023) Forecasting renewable energy generation with machine learning and deep learning: Current advances and future prospects. *Sustainability*, 15 (9), .

Chen, Y. & Xu, J. (2022) Solar and wind power data from the chinese state grid renewable energy generation forecasting competition. *Scientific Data*, 9 (1), 577.

Cocchi, G., Galli, L., Galvan, G., Sciandrone, M., Cantù, M. & Tomaselli, G. (2018) Machine learning methods for short-term bid forecasting in the renewable energy market: A case study in italy. *Wind Energy*, 21 (5), 357–371.

da Silva Fonseca Jr, J. G., Oozeki, T., Takashima, T., Koshimizu, G., Uchida, Y. & Ogimoto, K. (2012) Use of support vector regression and numerically predicted cloudiness to forecast power output of a photovoltaic power plant in kitakyushu, japan. *Progress in Photovoltaics: Research and Applications*, 20 (7), 874–882.

Das, U. K., Tey, K. S., Seyedmahmoudian, M., Mekhilef, S., Idris, M. Y. I., Van Deventer, W., Horan, B. & Stojcevski, A. (2018) Forecasting of photovoltaic power generation and model optimization: A review. *Renewable and Sustainable Energy Reviews*, 81 912–928.

Diagne, M., David, M., Lauret, P., Boland, J. & Schmutz, N. (2013) Review of solar irradiance forecasting methods and a proposition for small-scale insular grids. *Renewable and Sustainable Energy Reviews*, 27 65–76.

Donadio, L., Fang, J. & Porté-Agel, F. (2021) Numerical weather prediction and artificial neural network coupling for wind energy forecast. *Energies*, 14 (2), 338.

Dong, Z., Yang, D., Reindl, T. & Walsh, W. M. (2015) A novel hybrid approach based on self-organizing maps, support vector regression and particle swarm optimization to forecast solar irradiance. *Energy*, 82 570–577.

Dou, Y., Tan, S. & Xie, D. (2023a) Comparison of machine learning and statistical methods in the field of renewable energy power generation forecasting: A mini review. *Frontiers in Energy Research*, 11 1218603.

Dou, Y., Tan, S. & Xie, D. (2023b) Comparison of machine learning and statistical methods in the field of renewable energy power generation forecasting: A mini review. *Frontiers in Energy Research*, 11 1218603.

Dumitru, C., Gligor, A. & Enachescu, C. (2016) Solar photovoltaic energy production forecast using neural networks. *Procedia Technology*, 22 808–815.

Foley, A. M., Leahy, P. G., Marvuglia, A. & McKeogh, E. J. (2012) Current methods and advances in forecasting of wind power generation. *Renewable Energy*, 37 (1), 1–8.

Gastón, M., Pagola, I., Fernández-Peruchena, C. M., Ramírez, L. & Mallor, F. (2010) A new adaptive methodology of global-to-direct irradiance based on clustering and kernel machines techniques. *15th SolarPACES Conference*.

Gielen, D., Boshell, F., Saygin, D., Bazilian, M. D., Wagner, N. & Gorini, R. (2019) The role of renewable energy in the global energy transformation. *Energy Strategy Reviews*, 24 38–50.

Hewamalage, H., Bergmeir, C. & Bandara, K. (2021) Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37 (1), 388–427.

Huertas-Tato, J., Aler, R., Galván, I. M., Rodríguez-Benítez, F. J., Arbizu-Barrena, C. & Pozo-Vázquez, D. (2020) A short-term solar radiation forecasting system for the iberian peninsula. part 2: Model blending approaches based on machine learning. *Solar Energy*, 195 685–696.

Izgi, E., Öztopal, A., Yerli, B., Kaymak, M. K. & Şahin, A. D. (2012) Short–mid-term solar power prediction by using artificial neural networks. *Solar Energy*, 86 (2), 725–733.

Kane, M. J., Price, N., Scotch, M. & Rabinowitz, P. (2014) Comparison of ARIMA and random forest time series models for prediction of avian influenza H5N1 outbreaks. *BMC Bioinformatics*, 15 1–9.

Kassa, Y., Zhang, J. H., Zheng, D. H. & Wei, D. (2016) Short term wind power prediction using ANFIS. *2016 IEEE international conference on power and renewable energy (ICPRE)*. IEEE.

Kaufmann, W. (1981) The handbook of artificial intelligence.

Kazemzadeh, M., Amjadian, A. & Amraee, T. (2020) A hybrid data mining driven algorithm for long term electric peak load and energy demand forecasting. *Energy*, 204 117948.

Kumar, J. C. R., Kumar, D. V. & Majid, M. A. (2019) Wind energy programme in india. *Energy & Environment*, 30 (7), 1135–1189.

Li, Y., Su, Y. & Shu, L. (2014) An ARMAX model for forecasting the power output of a grid connected photovoltaic system. *Renewable Energy*, 66 78–89.

Lim, S., Huh, J., Hong, S., Park, C. & Kim, J. (2022) Solar power forecasting using CNN-LSTM hybrid model. *Energies*, 15 (21), 8233.

Lin, Z. & Liu, X. (2020) Wind power forecasting of an offshore wind turbine based on high-frequency SCADA data and deep learning neural network. *Energy*, 201 117693.

M. Elsaraiti & A. Merabet (2022) Solar power forecasting using deep learning techniques. *IEEE Access*, 10 31692–31698.

Mason, K., Duggan, J. & Howley, E. (2018) Forecasting energy demand, wind generation and carbon dioxide emissions in ireland using evolutionary neural networks. *Energy*, 155 705–720.

Methaprayoon, K., Lee, W. J., Didsayabutra, P., Liao, J. & Ross, R. (2003) Neural network-based short term load forecasting for unit commitment scheduling. *IEEE Technical Conference on Industrial and Commercial Power Systems*, 2003. IEEE.

Nazir, M. S., Alturise, F., Alshmrany, S., Nazir, H. M. J., Bilal, M., Abdalla, A. N., Sanjeevikumar, P. & M. Ali, Z. (2020a) Wind generation forecasting methods and proliferation of artificial neural network: A review of five years research trend. *Sustainability*, 12 (9), 3778.

Nazir, M. S., Alturise, F., Alshmrany, S., Nazir, H. M. J., Bilal, M., Abdalla, A. N., Sanjeevikumar, P. & M. Ali, Z. (2020b) Wind generation forecasting methods and proliferation of artificial neural network: A review of five years research trend. *Sustainability*, 12 (9), 3778.

Olatomiwa, L., Mekhilef, S., Shamshirband, S., Mohammadi, K., Petković, D. & Sudheer, C. (2015) A support vector machine–firefly algorithm-based model for global solar radiation prediction. *Solar Energy*, 115 632–644.

Rahman, M. M., Shakeri, M., Tiong, S. K., Khatun, F., Amin, N., Pasupuleti, J. & Hasan, M. K. (2021) Prospective methodologies in hybrid renewable energy systems for energy prediction using artificial neural networks. *Sustainability*, 13 (4), 2393.

Raza, M. Q. & Khosravi, A. (2015) A review on artificial intelligence-based load demand forecasting techniques for smart grid and buildings. *Renewable and Sustainable Energy Reviews*, 50 1352–1372.

Rhouma, A. & Yahia Said (2023a) Solar energy forecasting based on complex valued auto-encoder and recurrent neural network. *International Journal of Advanced Computer Science and Applications*, 14 (4), .

Rhouma, A. & Yahia Said (2023b) Solar energy forecasting based on complex valued auto-encoder and recurrent neural network. *International Journal of Advanced Computer Science and Applications*, 14 (4), .

Ruzic, S., Vuckovic, A. & Nikolic, N. (2003) Weather sensitive method for short term load forecasting in electric power utility of serbia. *IEEE Transactions on Power Systems*, 18 (4), 1581–1586.

S. Rahman & O. Hazim (1993) A generalized knowledge-based short-term load-forecasting technique. *IEEE Transactions on Power Systems*, 8 (2), 508–514.

Shi, J., Lee, W., Liu, Y., Yang, Y. & Wang, P. (2012) Forecasting power output of photovoltaic systems based on weather classification and support vector machines. *IEEE Transactions on Industry Applications*, 48 (3), 1064–1069.

Son, J., Park, Y., Lee, J. & Kim, H. (2018) Sensorless PV power forecasting in grid-connected buildings through deep learning. *Sensors*, 18 (8), 2529.

Wang, H., Lei, Z., Zhang, X., Zhou, B. & Peng, J. (2019) A review of deep learning for renewable energy forecasting. *Energy Conversion and Management*, 198 111799.

Wang, K., Wang, J., Zeng, B. & Lu, H. (2022) An integrated power load point-interval forecasting system based on information entropy and multi-objective optimization. *Applied Energy*, 314 118938.

Yang, G., Wang, Y. & Li, X. (2020) Prediction of the NO<sub>x</sub> emissions from thermal power plant using long-short term memory neural network. *Energy*, 192 116597.

Yang, H., Huang, C., Huang, Y. & Pai, Y. (2014) A weather-based hybrid method for 1-day ahead hourly forecasting of PV power output. *IEEE Transactions on Sustainable Energy*, 5 (3), 917–926.

## Appendix 1 Model Results

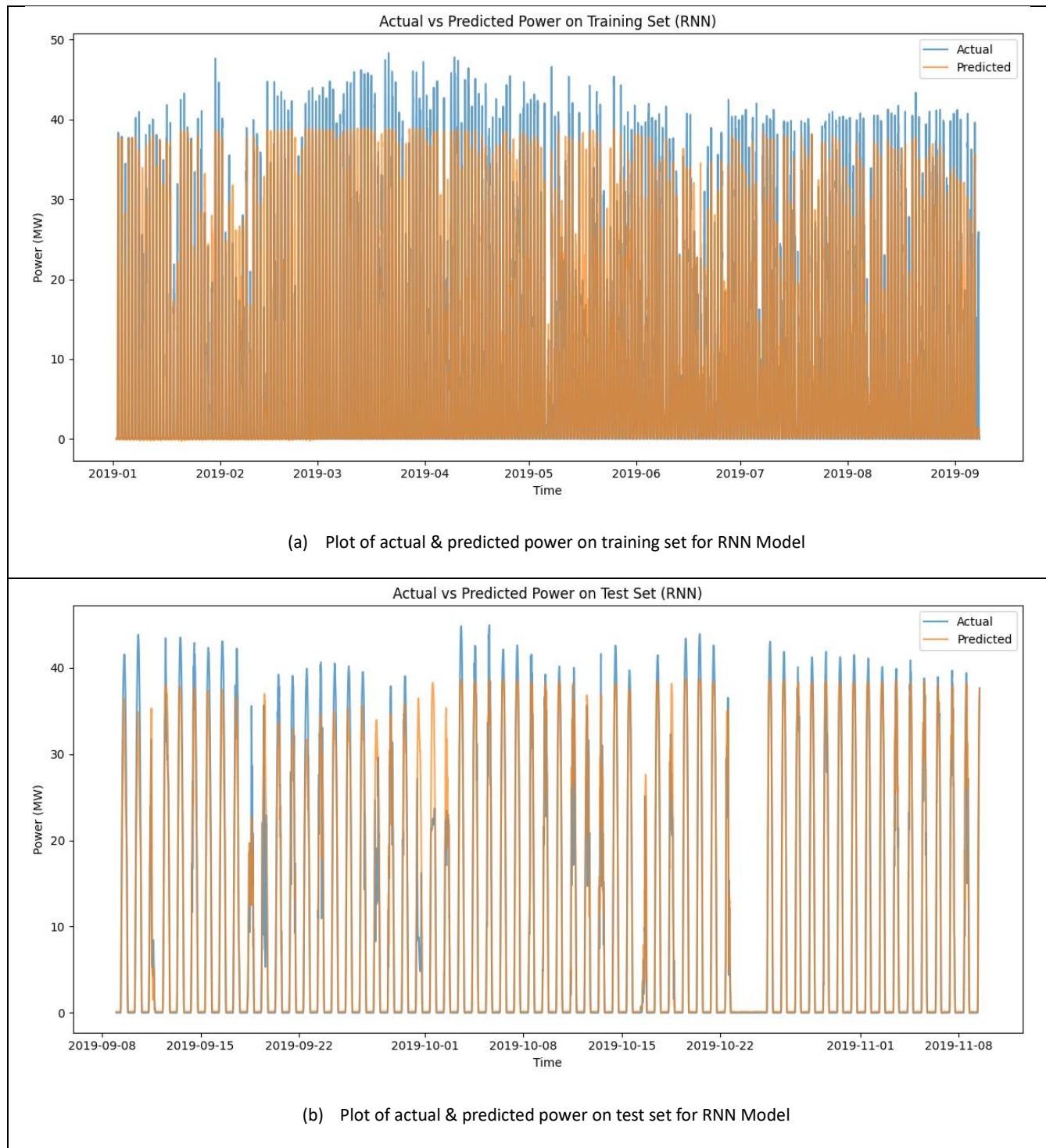
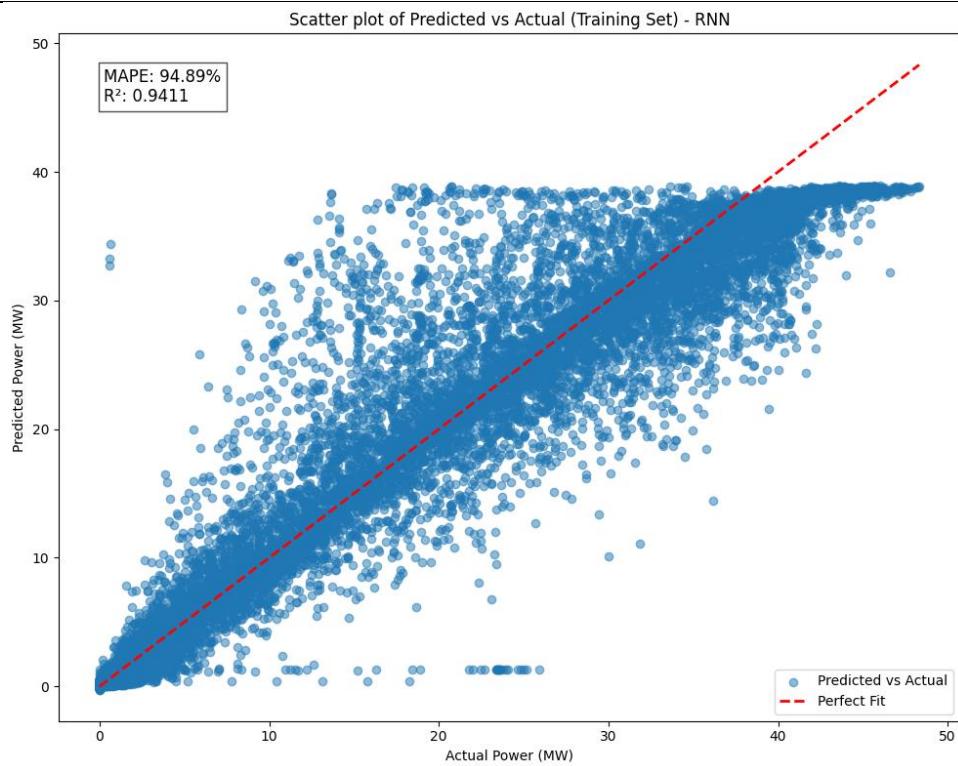
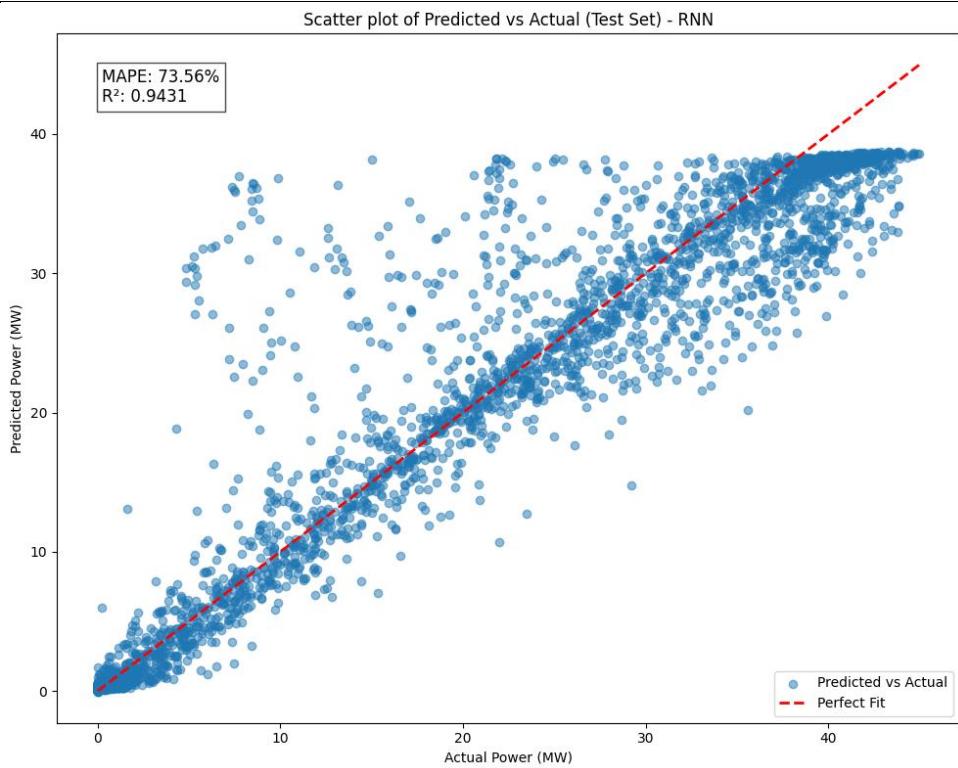


Figure A1. 1 Actual & predicted power on training and test set for RNN model



(a) Scatter plot of predicted vs actual on training set for RNN model



(b) Scatter plot of predicted vs actual on tests set for RNN model

Figure A1. 2 Scatter plot of predicted vs actual on training and test set for RNN model

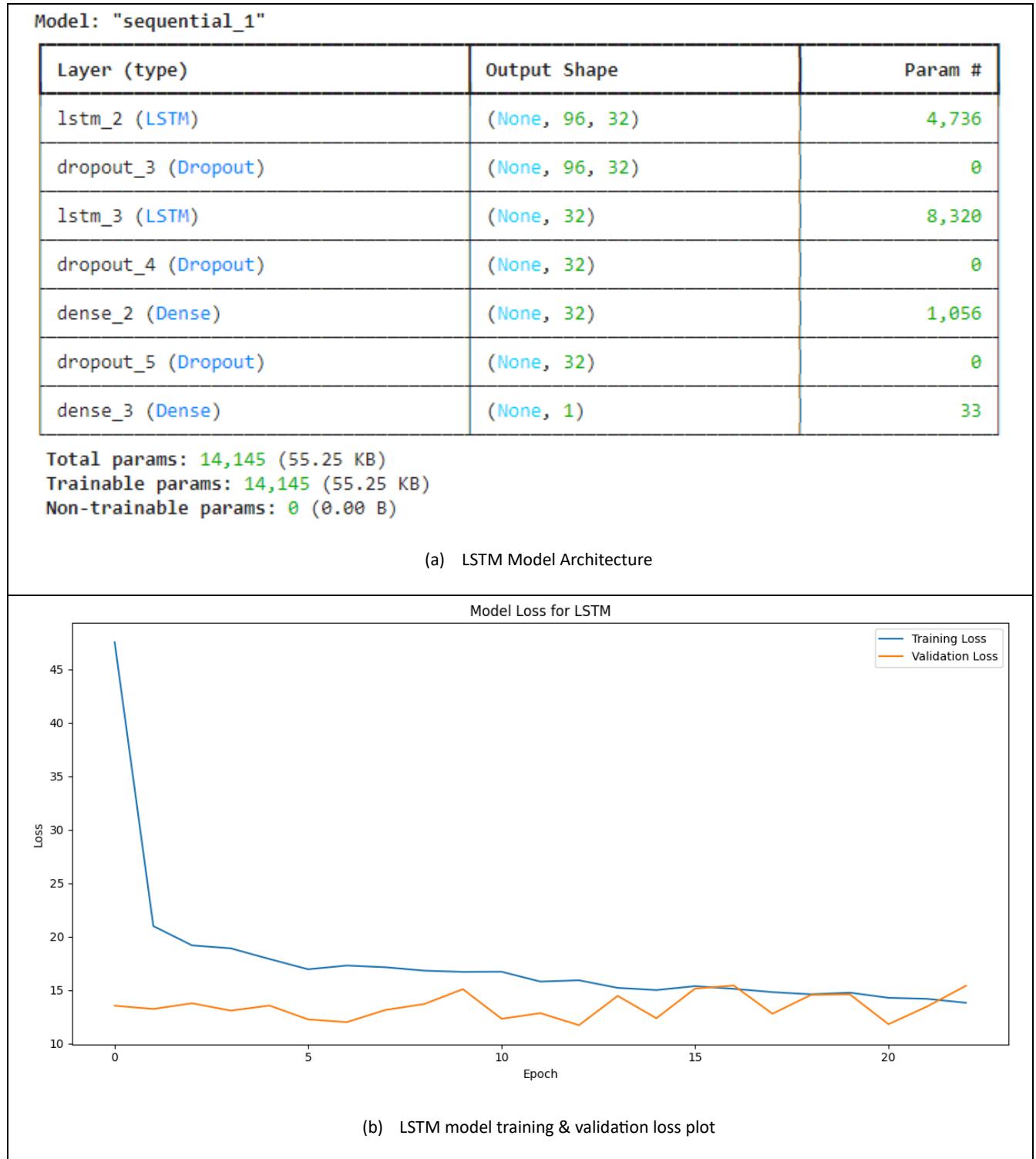


Figure A1. 3(a) LSTM model summary architecture (b) LSTM model training & validation loss plot

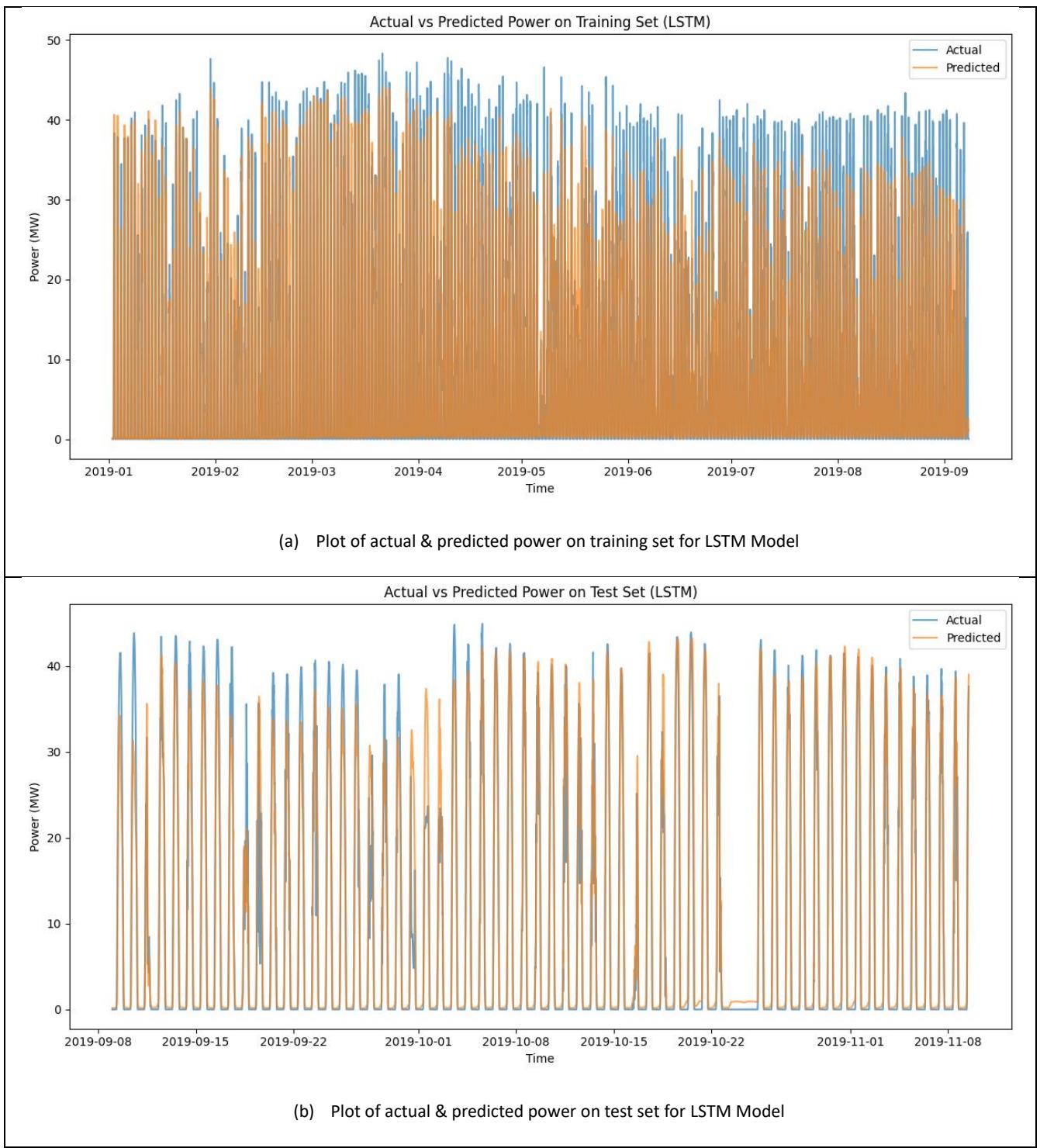
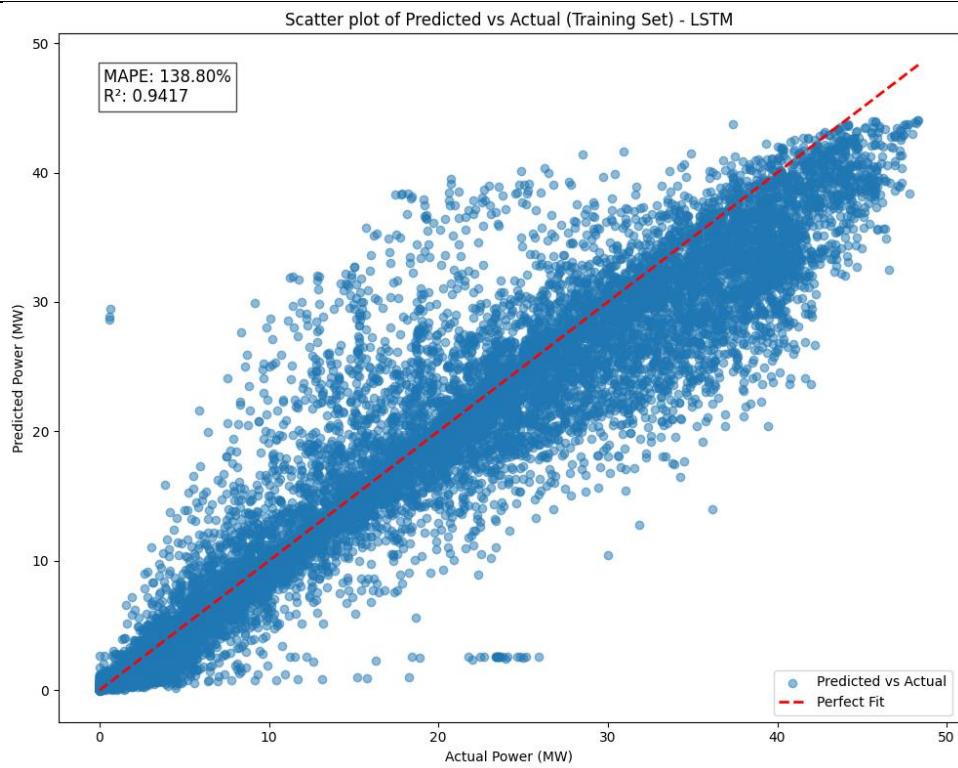
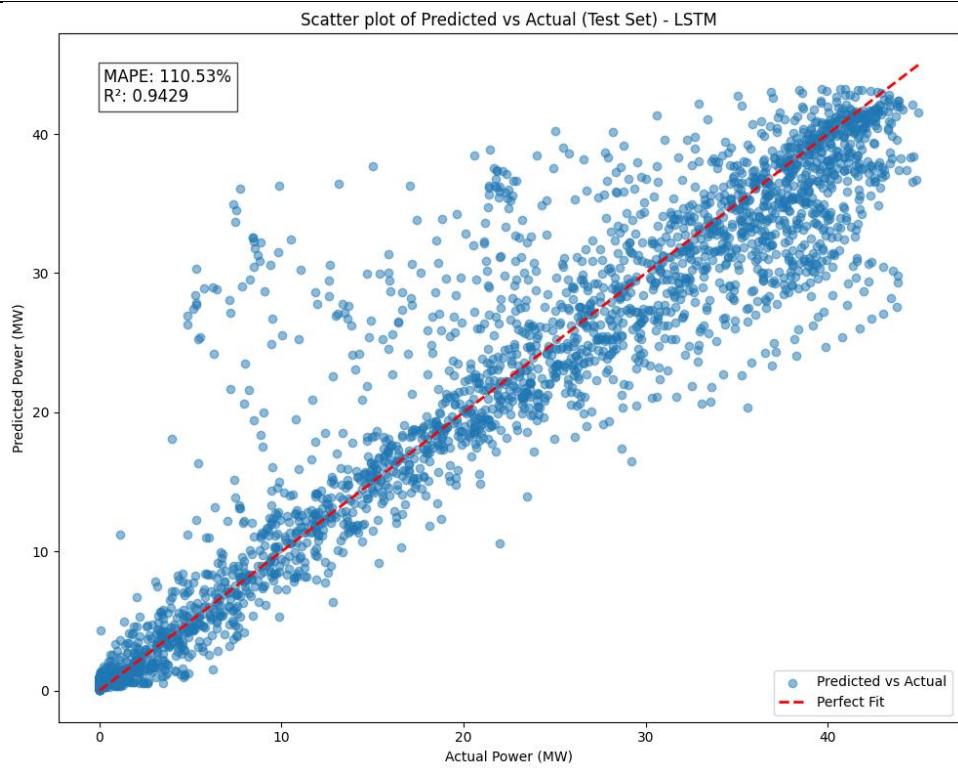


Figure A1. 4 Actual & predicted power on training and test set for LSTM model



(a) Scatter plot of predicted vs actual on training set for LSTM model



(b) Scatter plot of predicted vs actual on test set for LSTM model

Figure A1. 5 Scatter plot of predicted vs actual power on training and test set for LSTM model

Model: "sequential\_1"

Layer (type)	Output Shape	Param #
gru_2 (GRU)	(None, 96, 64)	13,440
dropout_4 (Dropout)	(None, 96, 64)	0
gru_3 (GRU)	(None, 64)	24,960
dropout_5 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 32)	2,080
dropout_6 (Dropout)	(None, 32)	0
dense_4 (Dense)	(None, 1)	33

Total params: 40,513 (158.25 KB)

Trainable params: 40,513 (158.25 KB)

Non-trainable params: 0 (0.00 B)

(a) GRU model summary architecture

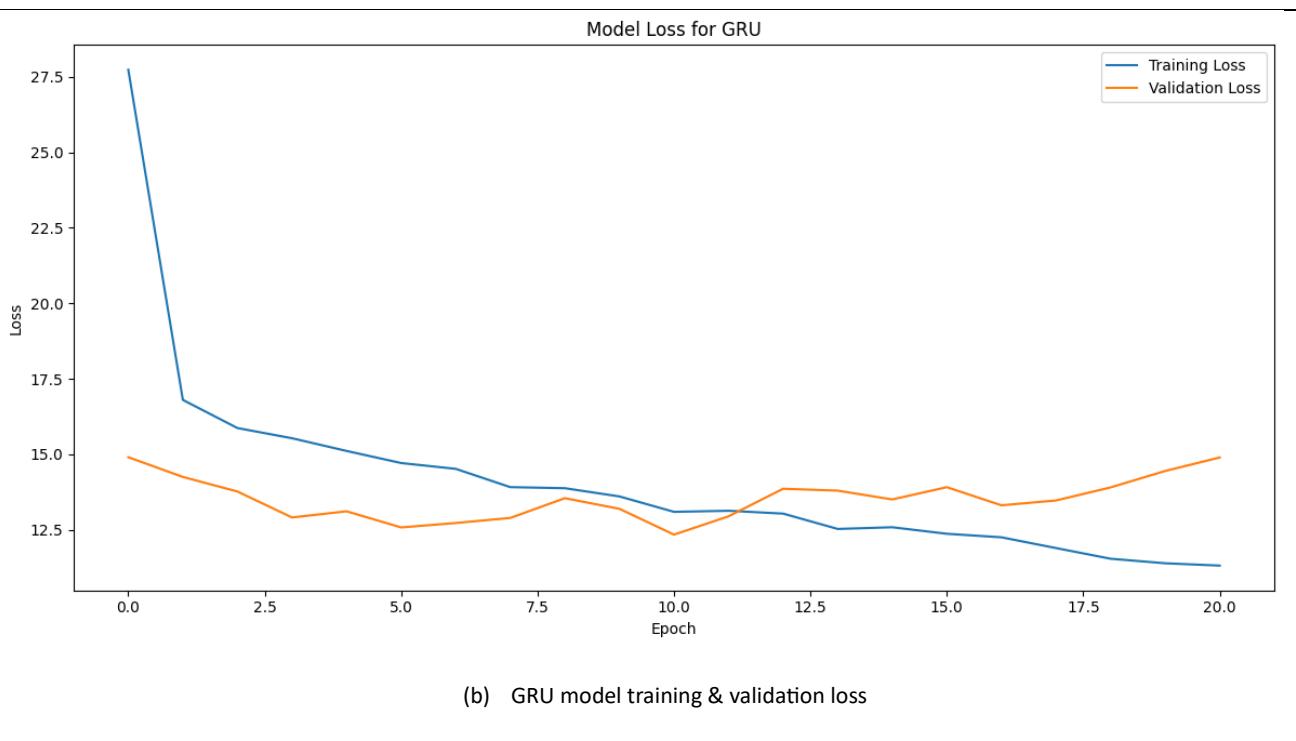


Figure A1. 6 (a) GRU model summary architecture (b) GRU model training & validation loss plot

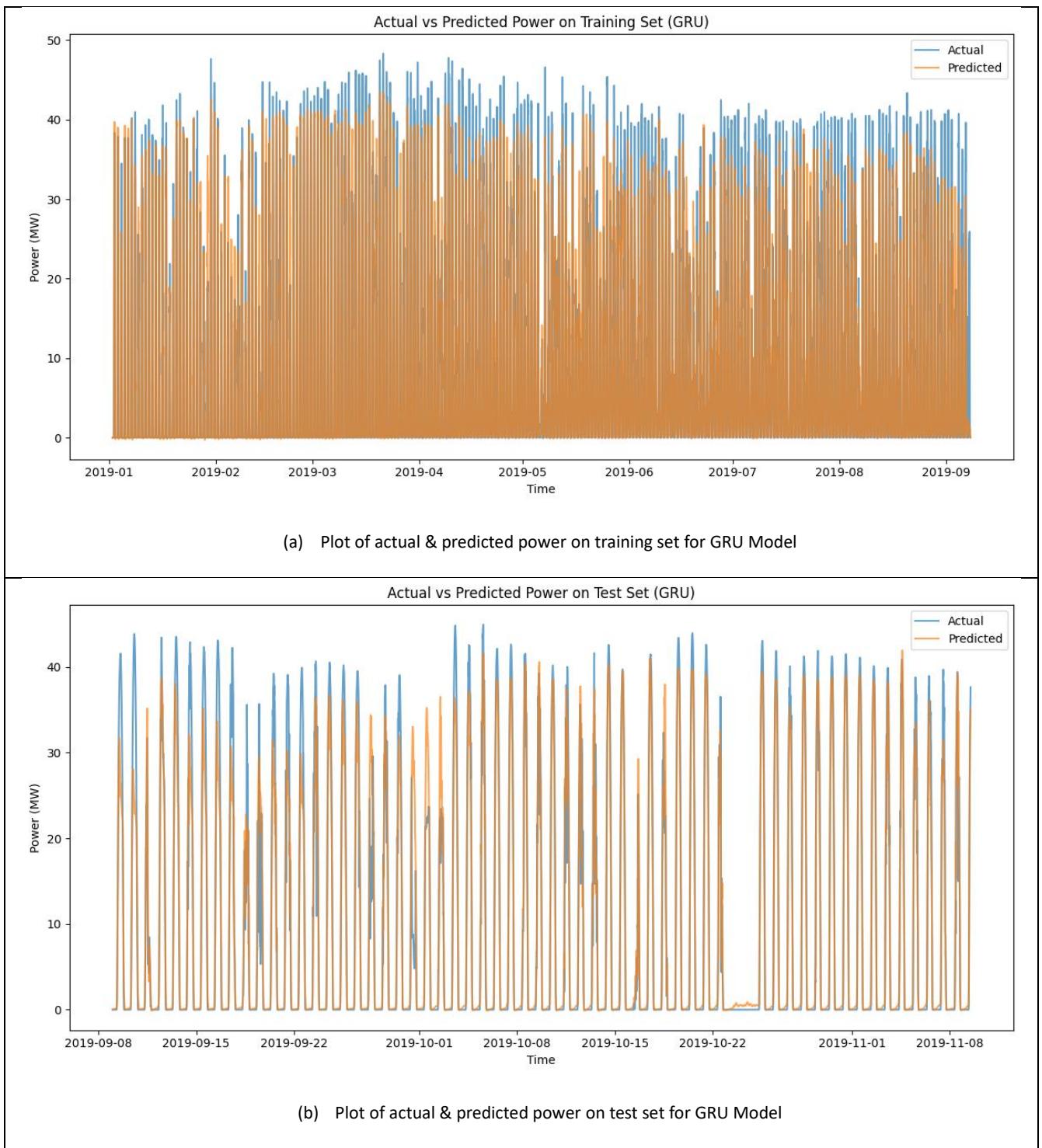
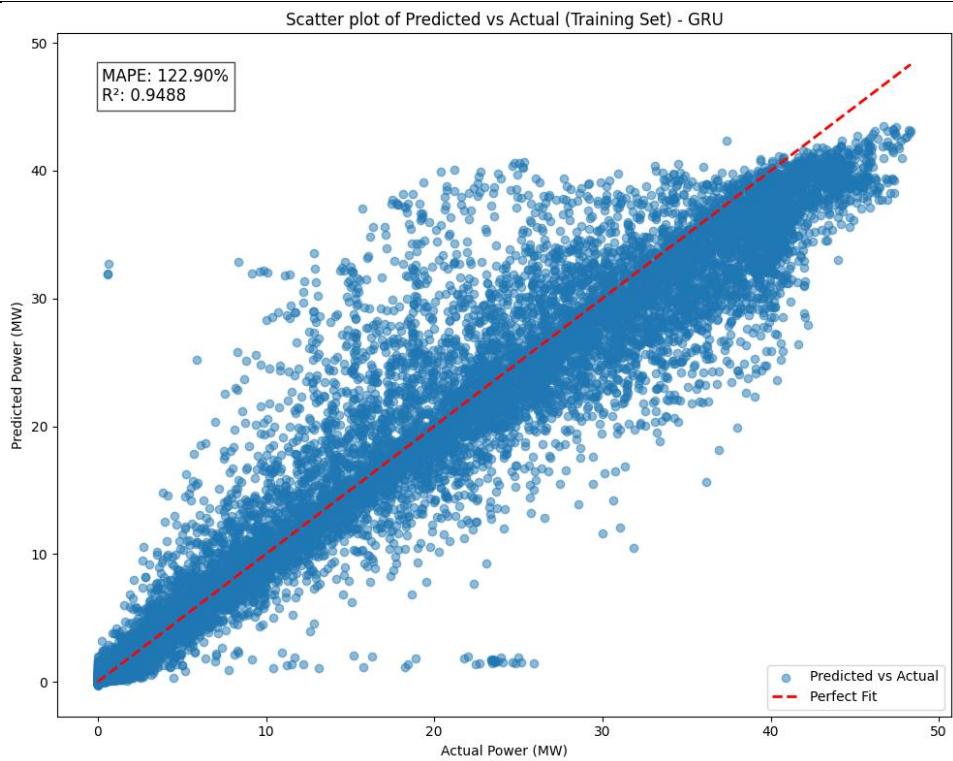
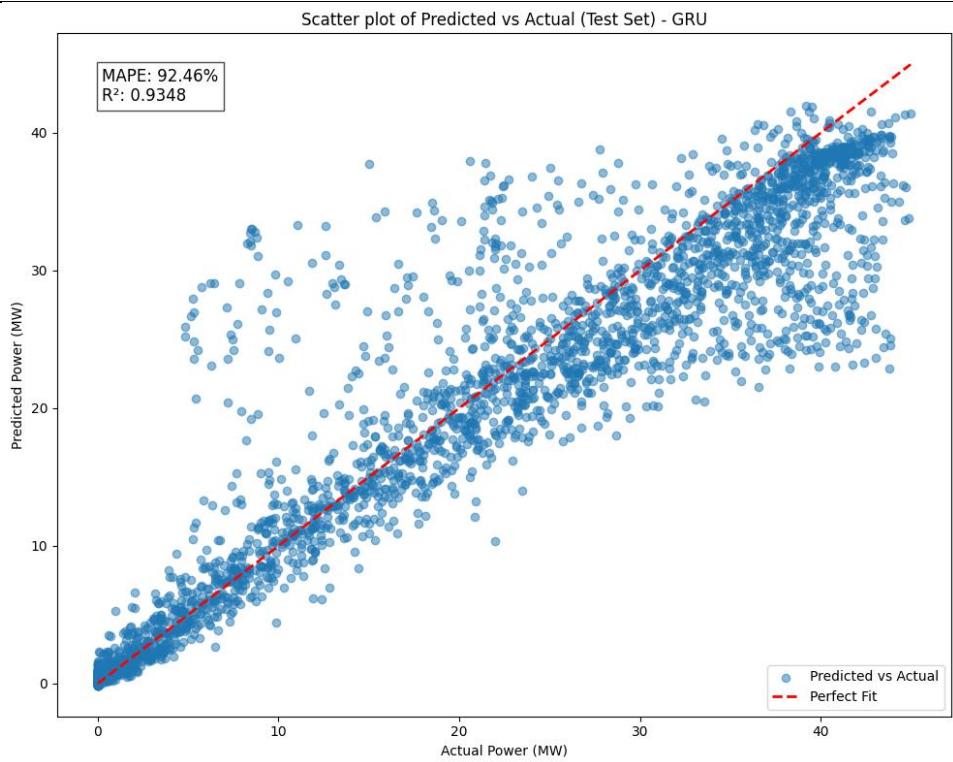


Figure A1. 7 Actual & predicted power on training and test set for GRU model



(a) Scatter plot of predicted vs actual power on training set for GRU model



(b) Scatter plot of predicted vs actual power on training set for GRU model

Figure A1. 8 Scatter plot of predicted vs actual power son training and test set for GRU model

Model: "sequential\_1"

Layer (type)	Output Shape	Param #
conv1d_2 (Conv1D)	(None, 94, 32)	416
dropout_3 (Dropout)	(None, 94, 32)	0
max_pooling1d_2 (MaxPooling1D)	(None, 47, 32)	0
flatten_1 (Flatten)	(None, 1504)	0
dense_2 (Dense)	(None, 32)	48,160
dropout_4 (Dropout)	(None, 32)	0
dense_3 (Dense)	(None, 16)	528
dropout_5 (Dropout)	(None, 16)	0
dense_4 (Dense)	(None, 1)	17

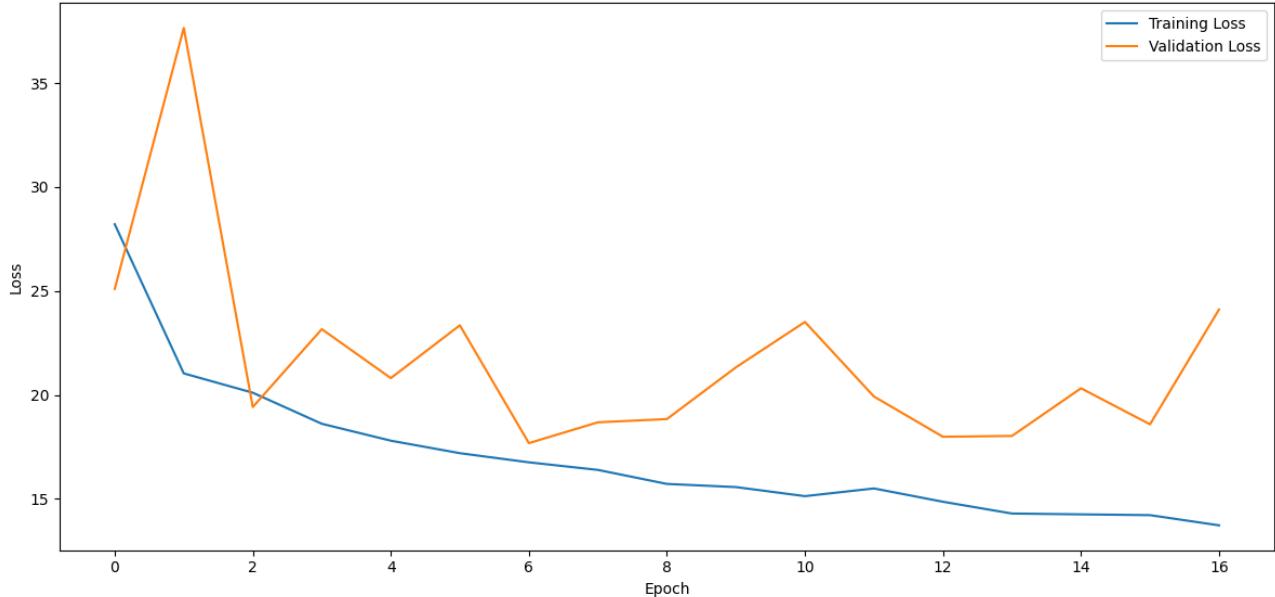
Total params: 49,121 (191.88 KB)

Trainable params: 49,121 (191.88 KB)

Non-trainable params: 0 (0.00 B)

(a) CNN model summary architecture

Model Loss for Conv1D



(b) CNN model training & validation loss

Figure A1. 9 (a) CNN model summary architecture (b) CNN model training & validation loss plot

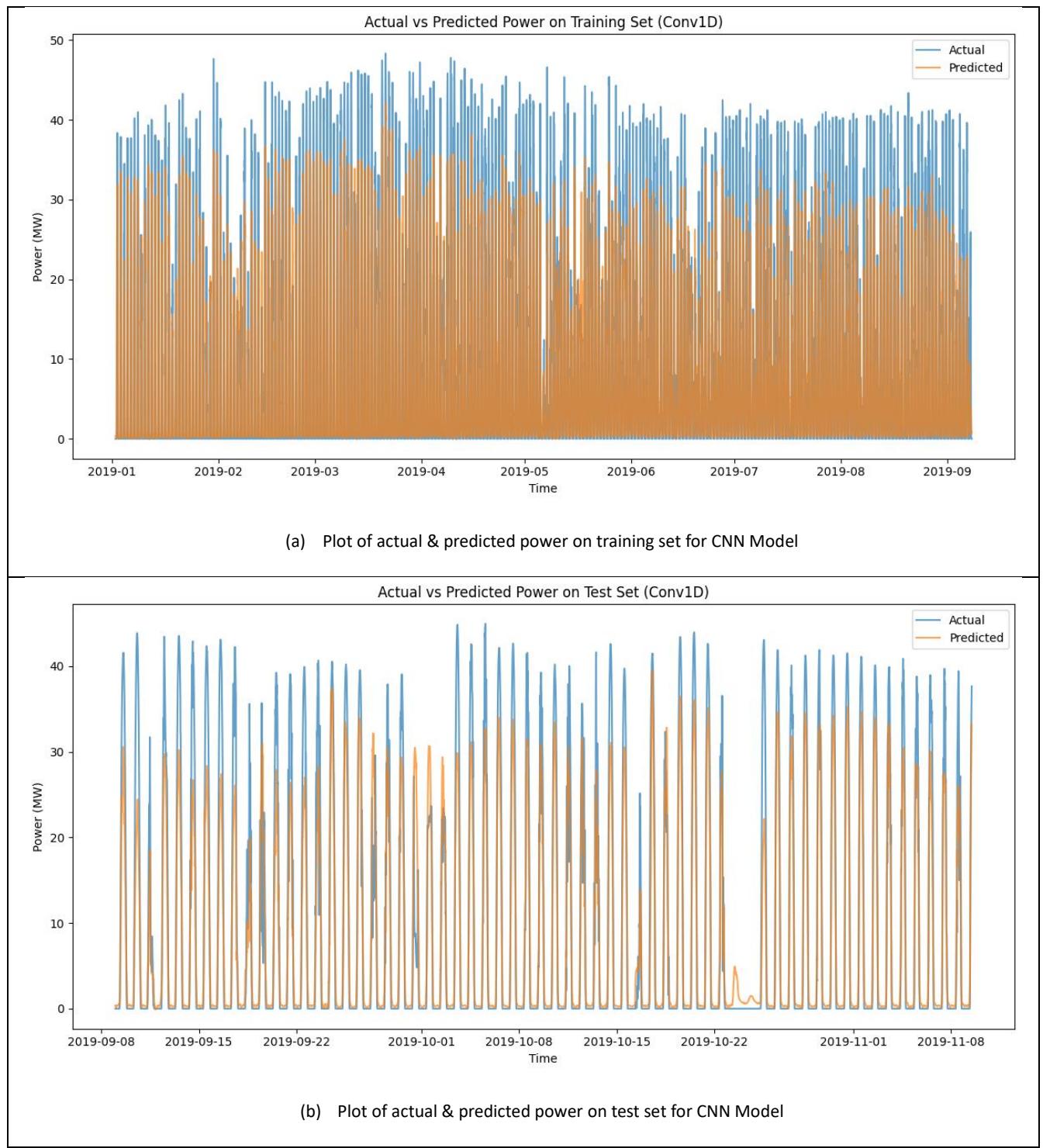
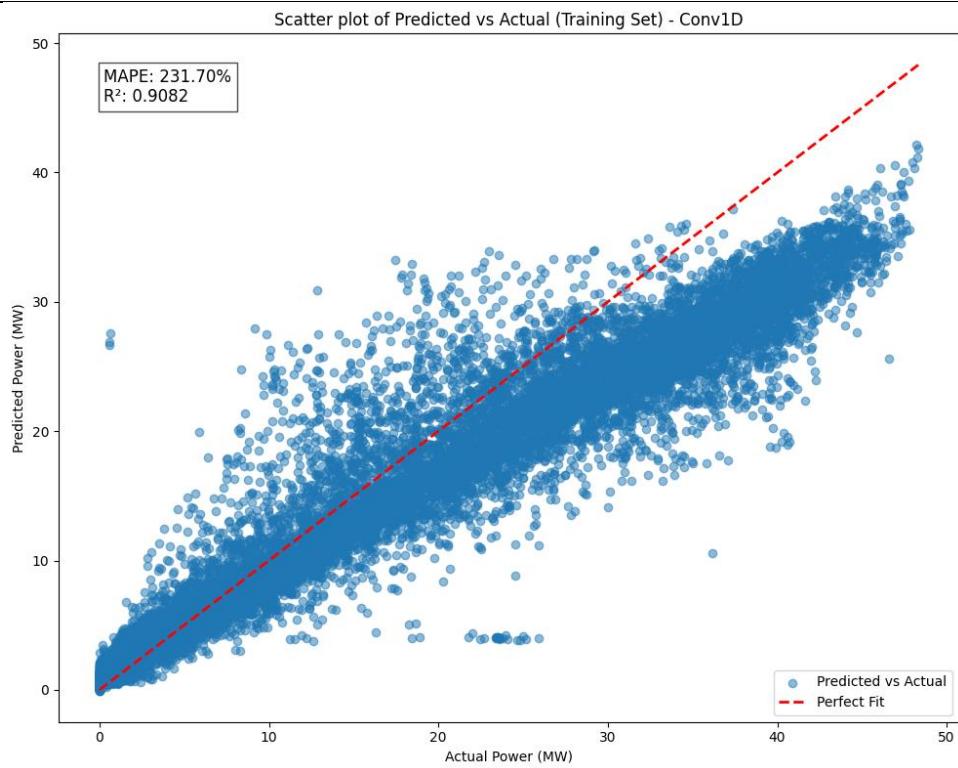
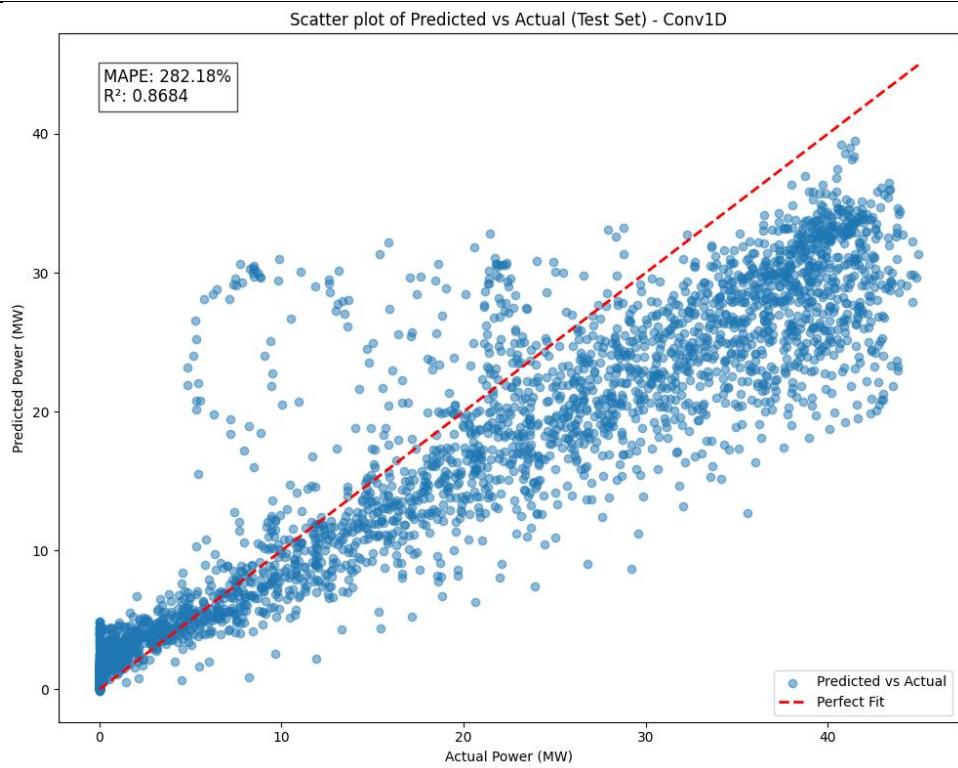


Figure A1. 10 Actual & predicted power on training and test set for CNN model



(a) Scatter plot of predicted vs actual power on training set for CNN model



(b) Scatter plot of predicted vs actual power on test set for CNN model

Figure A1. 11 Scatter plot of predicted vs actual power on training and test set for CNN model

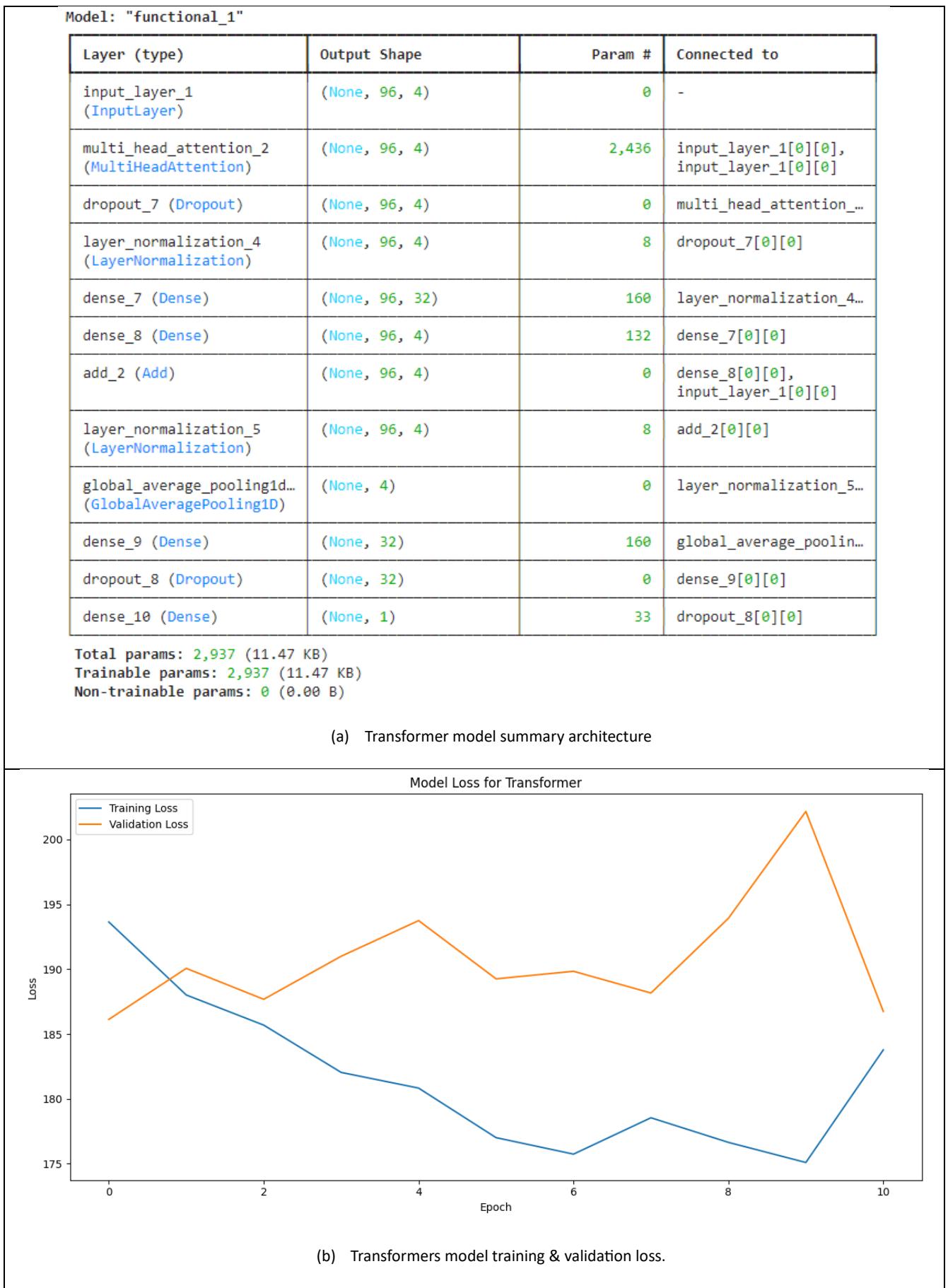


Figure A1. 12 (a) Transformers model summary architecture (b) model training & validation loss plot

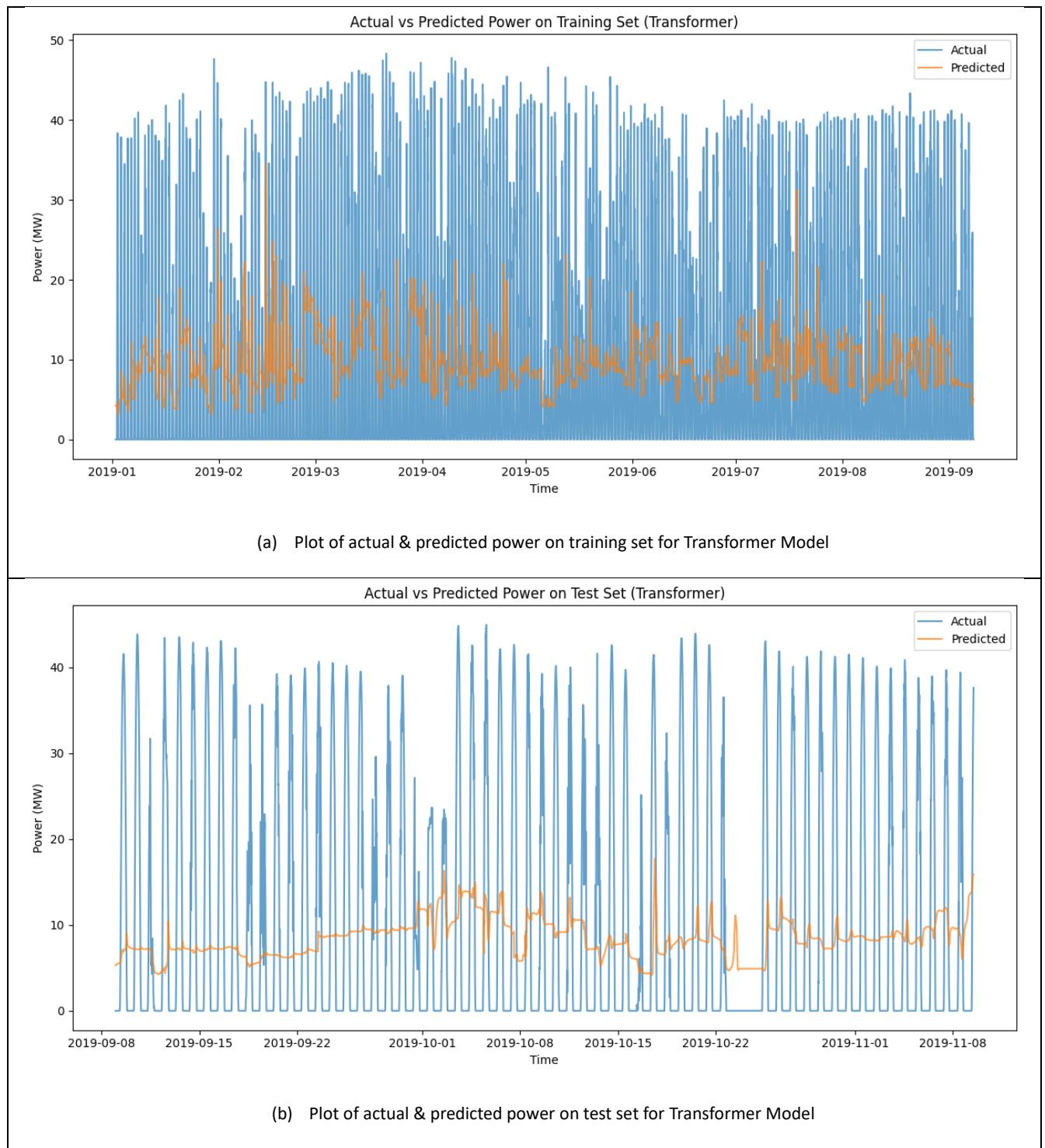
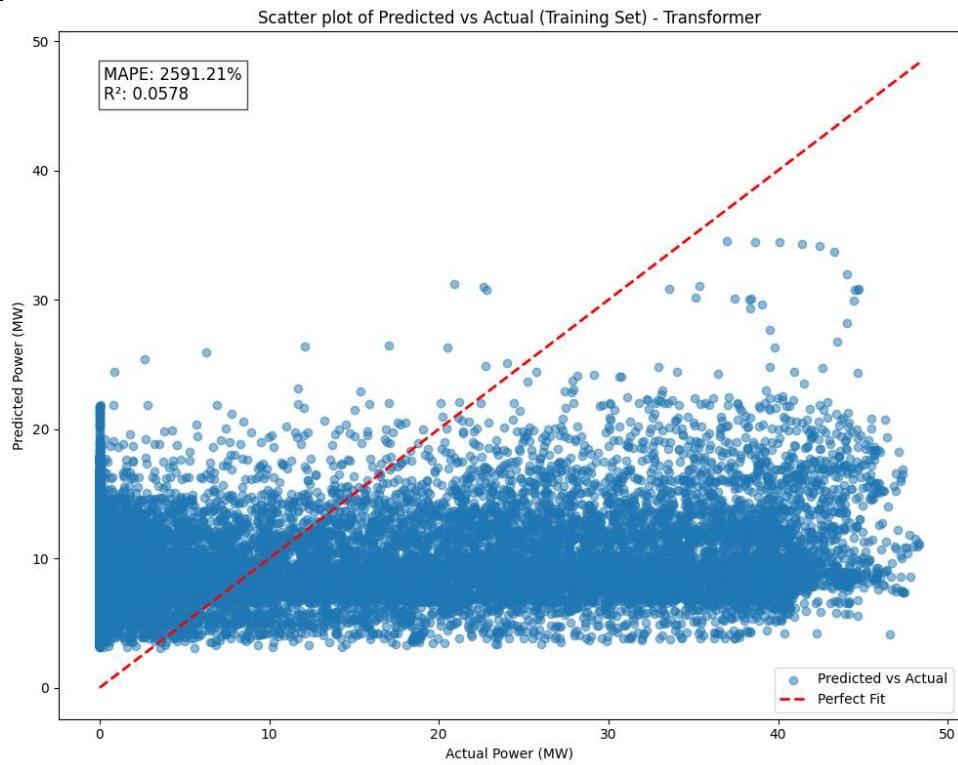
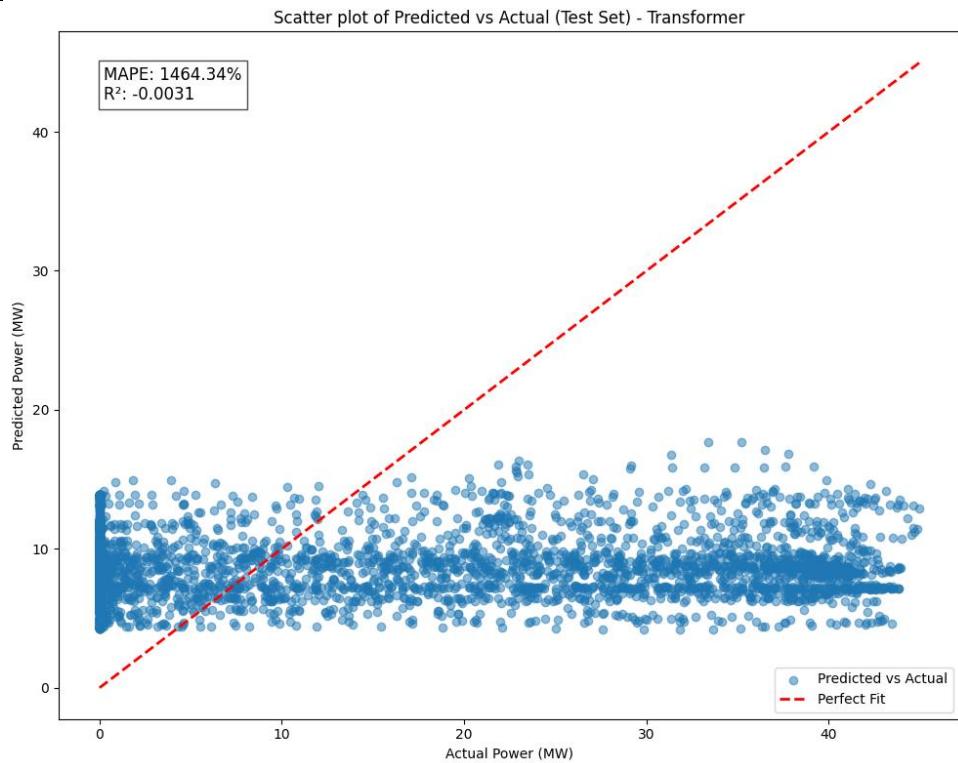


Figure A1. 13 Actual & predicted power on training & test set for transformers model



(a) Scatter plot of predicted vs actual power on training set for Transformers model



(b) Scatter plot of predicted vs actual power on test set for Transformers model

Figure A1. 14 Scatterplot of predicted vs actual power on training and test set for transformers model

Model: "functional\_1"

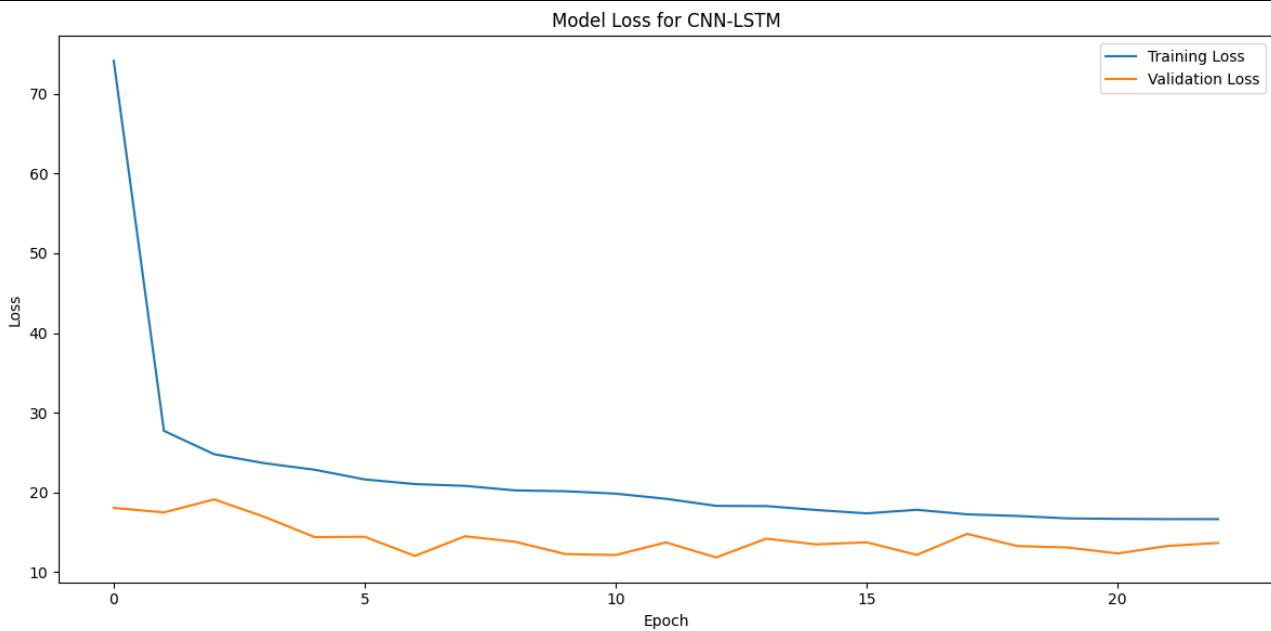
Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 96, 4)	0
conv1d_2 (Conv1D)	(None, 92, 32)	672
max_pooling1d_2 (MaxPooling1D)	(None, 46, 32)	0
dropout_5 (Dropout)	(None, 46, 32)	0
lstm_2 (LSTM)	(None, 46, 32)	8,320
dropout_6 (Dropout)	(None, 46, 32)	0
lstm_3 (LSTM)	(None, 32)	8,320
dropout_7 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 16)	528
dropout_8 (Dropout)	(None, 16)	0
dense_3 (Dense)	(None, 16)	272
dropout_9 (Dropout)	(None, 16)	0
dense_4 (Dense)	(None, 1)	17

Total params: 18,129 (70.82 KB)

Trainable params: 18,129 (70.82 KB)

Non-trainable params: 0 (0.00 B)

(a) CNN-LSTM model summary architecture



(b) CNN-LSTM model training & validation loss plot

Figure A1. 15(a) CNN-LSTM model summary architecture (b) model training & validation loss plot

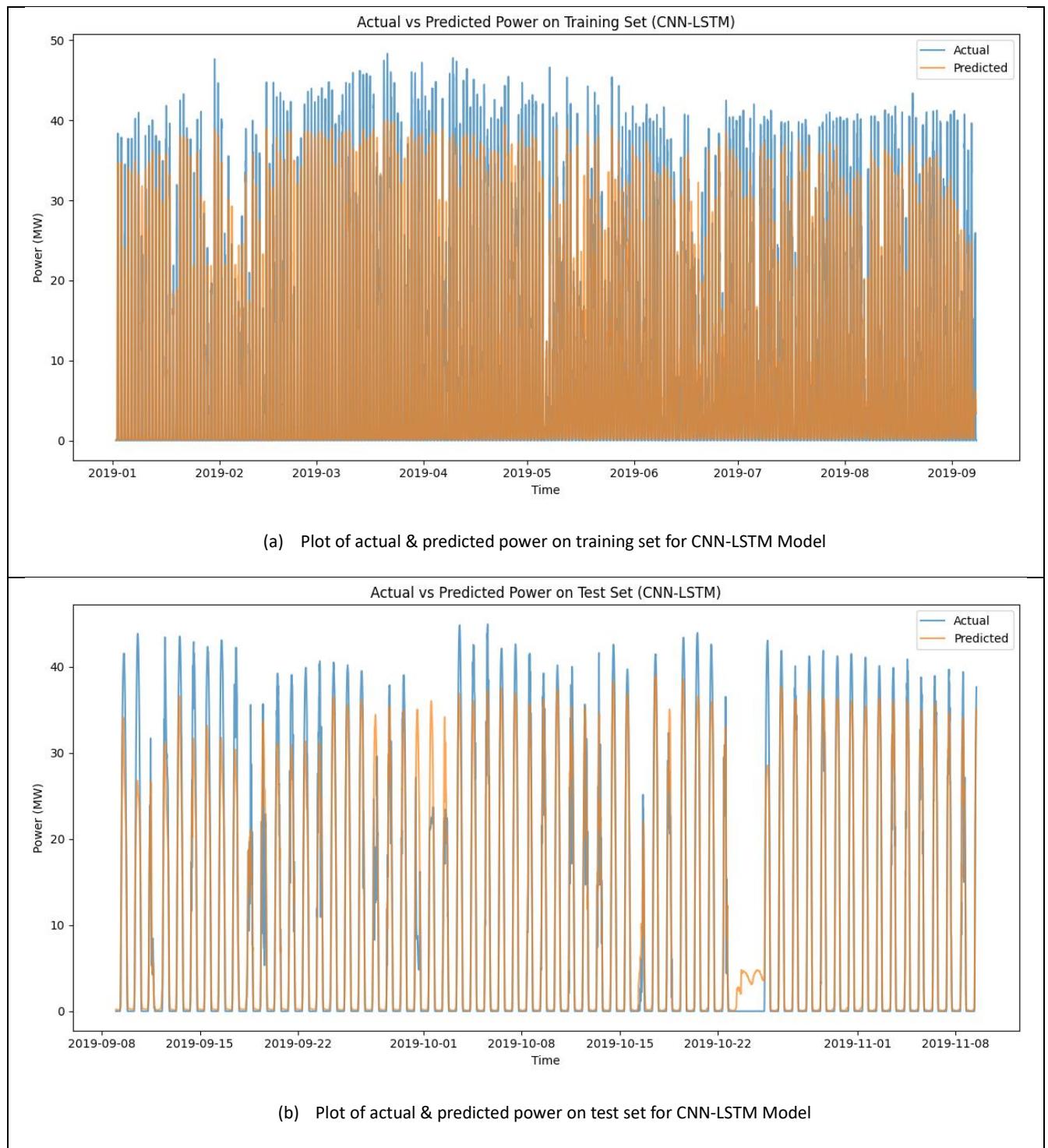


Figure A1. 16 Actual & predicted power on training & test set for CNN-LSTM model

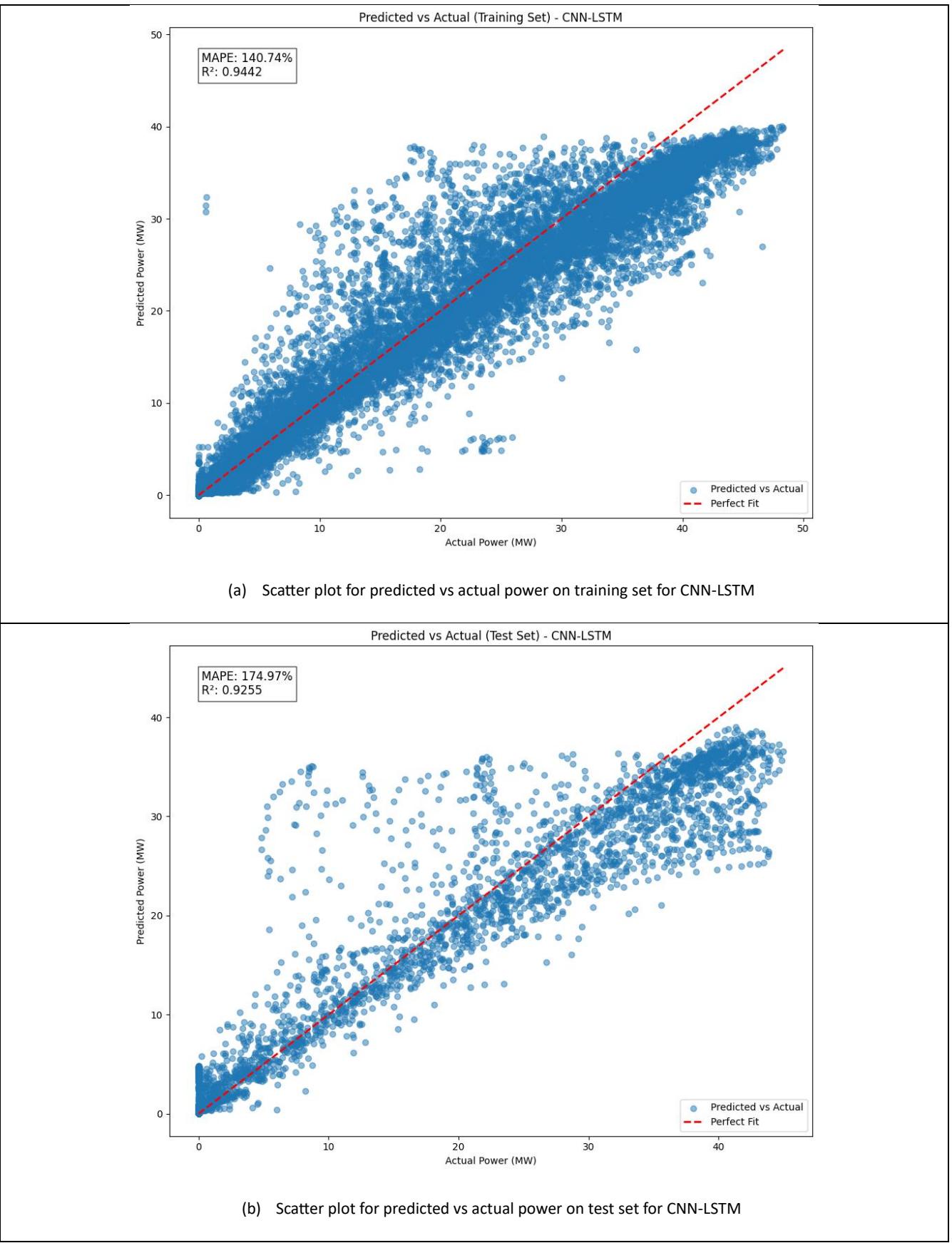


Figure A1. 17 Scatterplot of predicted vs actual power on training and test set for CNN-LSTM model

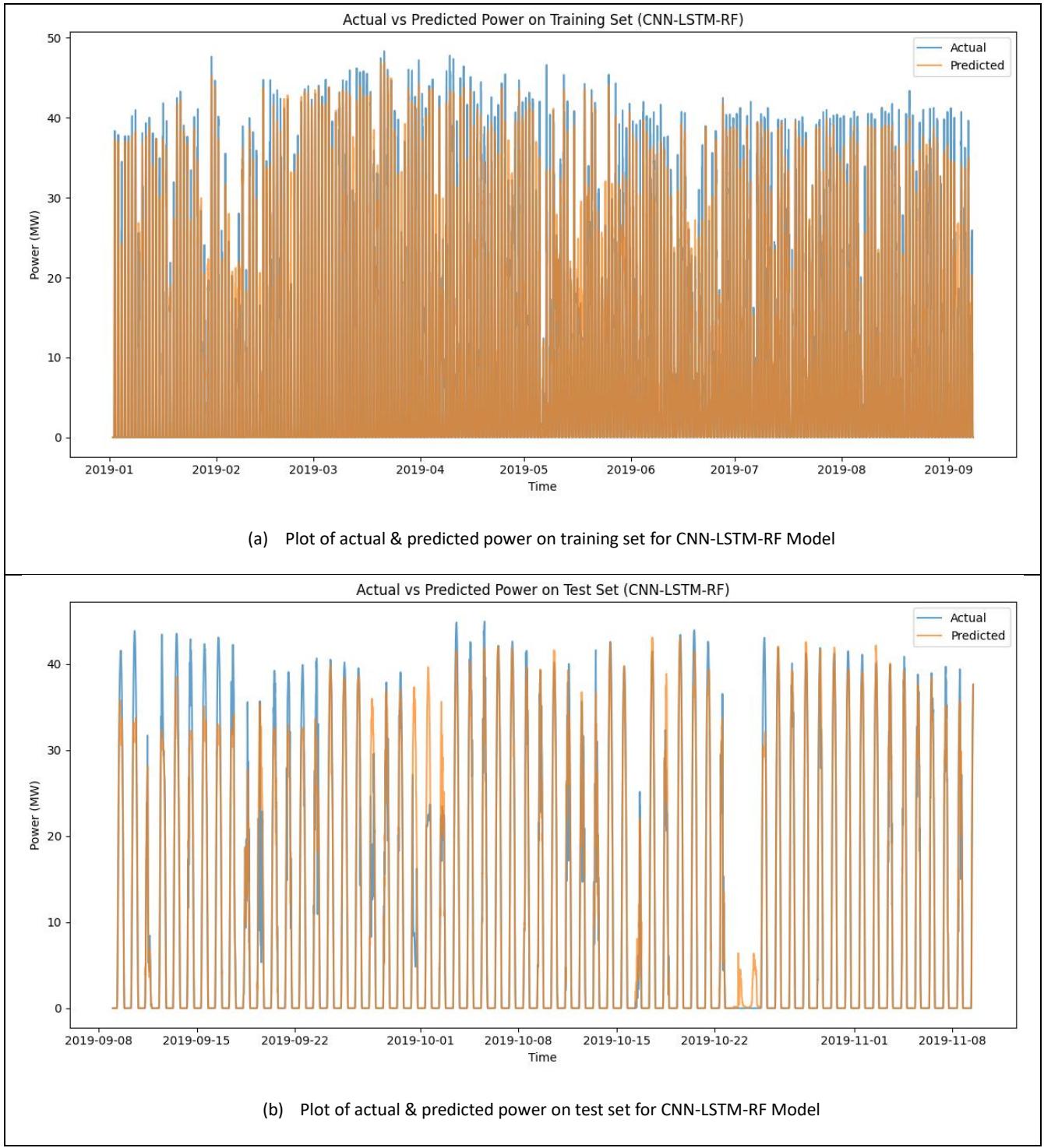


Figure A1. 18 Actual & predicted power on training & test set for CNN-LSTM-RF model

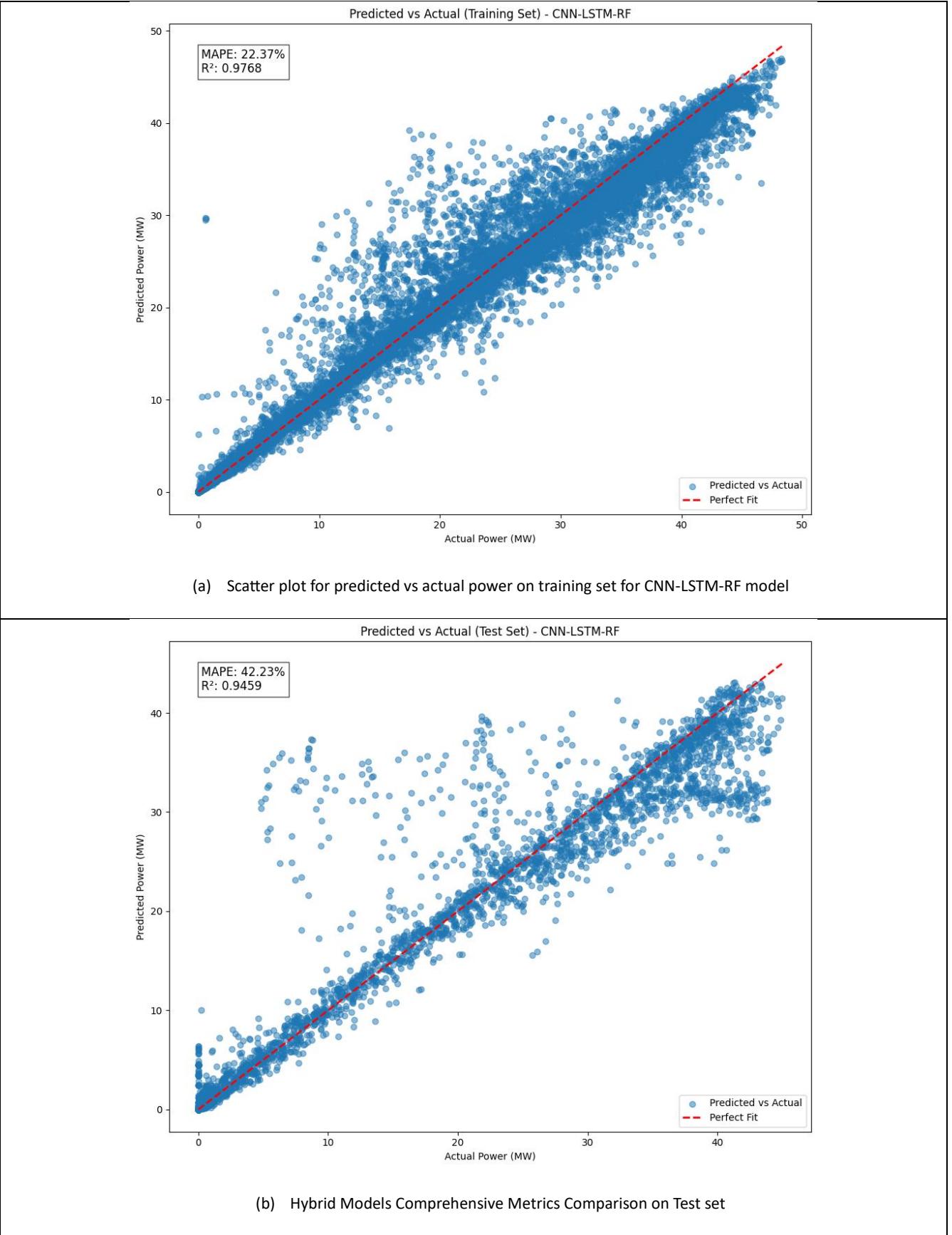


Figure A1. 19 Scatterplot of predicted vs actual power on training and test set for CNN-LSTM-RF model

Model: "functional\_1"

Layer (type)	Output Shape	Param #	Connected to
input_layer_1 (InputLayer)	(None, 96, 4)	0	-
conv1d_1 (Conv1D)	(None, 96, 16)	336	input_layer_1[0][0]
max_pooling1d_1 (MaxPooling1D)	(None, 48, 16)	0	conv1d_1[0][0]
conv1d_2 (Conv1D)	(None, 48, 16)	784	max_pooling1d_1[0][0]
max_pooling1d_2 (MaxPooling1D)	(None, 24, 16)	0	conv1d_2[0][0]
lstm_2 (LSTM)	(None, 24, 64)	20,736	max_pooling1d_2[0][0]
dropout_10 (Dropout)	(None, 24, 64)	0	lstm_2[0][0]
multi_head_attention_2 (MultiHeadAttention)	(None, 24, 64)	16,640	dropout_10[0][0], dropout_10[0][0]
dropout_12 (Dropout)	(None, 24, 64)	0	multi_head_attention_2[0][0]
add_4 (Add)	(None, 24, 64)	0	dropout_10[0][0], dropout_12[0][0]
layer_normalization_4 (LayerNormalization)	(None, 24, 64)	128	add_4[0][0]
dense_7 (Dense)	(None, 24, 64)	4,160	layer_normalization_4[0][0]
dense_8 (Dense)	(None, 24, 64)	4,160	dense_7[0][0]
dropout_13 (Dropout)	(None, 24, 64)	0	dense_8[0][0]
add_5 (Add)	(None, 24, 64)	0	layer_normalization_4[0][0], dropout_13[0][0]
layer_normalization_5 (LayerNormalization)	(None, 24, 64)	128	add_5[0][0]
multi_head_attention_3 (MultiHeadAttention)	(None, 24, 64)	16,640	layer_normalization_5[0][0], layer_normalization_5[0][0]
dropout_15 (Dropout)	(None, 24, 64)	0	multi_head_attention_3[0][0]
add_6 (Add)	(None, 24, 64)	0	layer_normalization_5[0][0], dropout_15[0][0]
layer_normalization_6 (LayerNormalization)	(None, 24, 64)	128	add_6[0][0]
dense_9 (Dense)	(None, 24, 64)	4,160	layer_normalization_6[0][0]
dense_10 (Dense)	(None, 24, 64)	4,160	dense_9[0][0]
dropout_16 (Dropout)	(None, 24, 64)	0	dense_10[0][0]
add_7 (Add)	(None, 24, 64)	0	layer_normalization_6[0][0], dropout_16[0][0]
layer_normalization_7 (LayerNormalization)	(None, 24, 64)	128	add_7[0][0]
global_average_pooling1d_1 (GlobalAveragePooling1D)	(None, 64)	0	layer_normalization_7[0][0]
dense_11 (Dense)	(None, 64)	4,160	global_average_pooling1d_1[0][0]
dropout_17 (Dropout)	(None, 64)	0	dense_11[0][0]
dense_12 (Dense)	(None, 64)	4,160	dropout_17[0][0]
dropout_18 (Dropout)	(None, 64)	0	dense_12[0][0]
dense_13 (Dense)	(None, 1)	65	dropout_18[0][0]

Total params: 80,673 (315.13 KB)  
 Trainable params: 80,673 (315.13 KB)  
 Non-trainable params: 0 (0.00 B)

(a) CNN-LSTM-TR model summary architecture

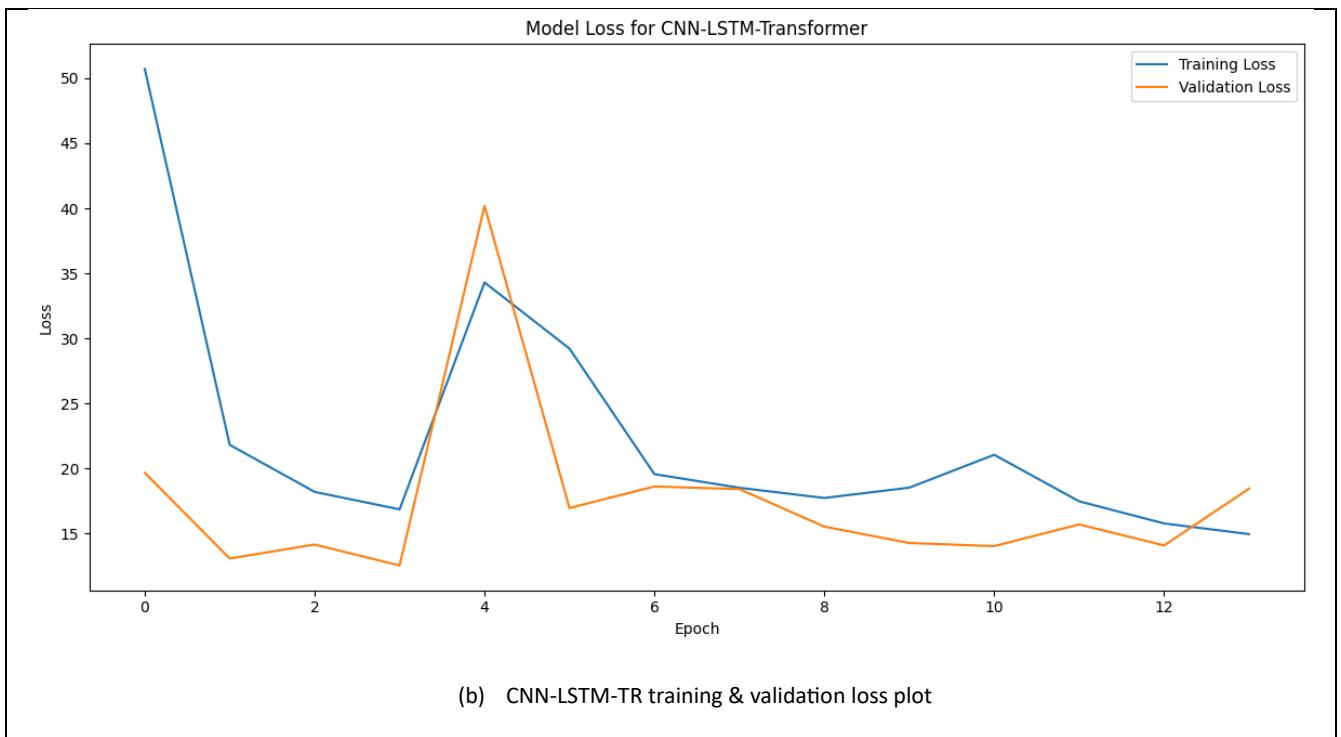


Figure A1. 20 (a) CNN-LSTM-TR model summary architecture (b) model training & validation loss plot

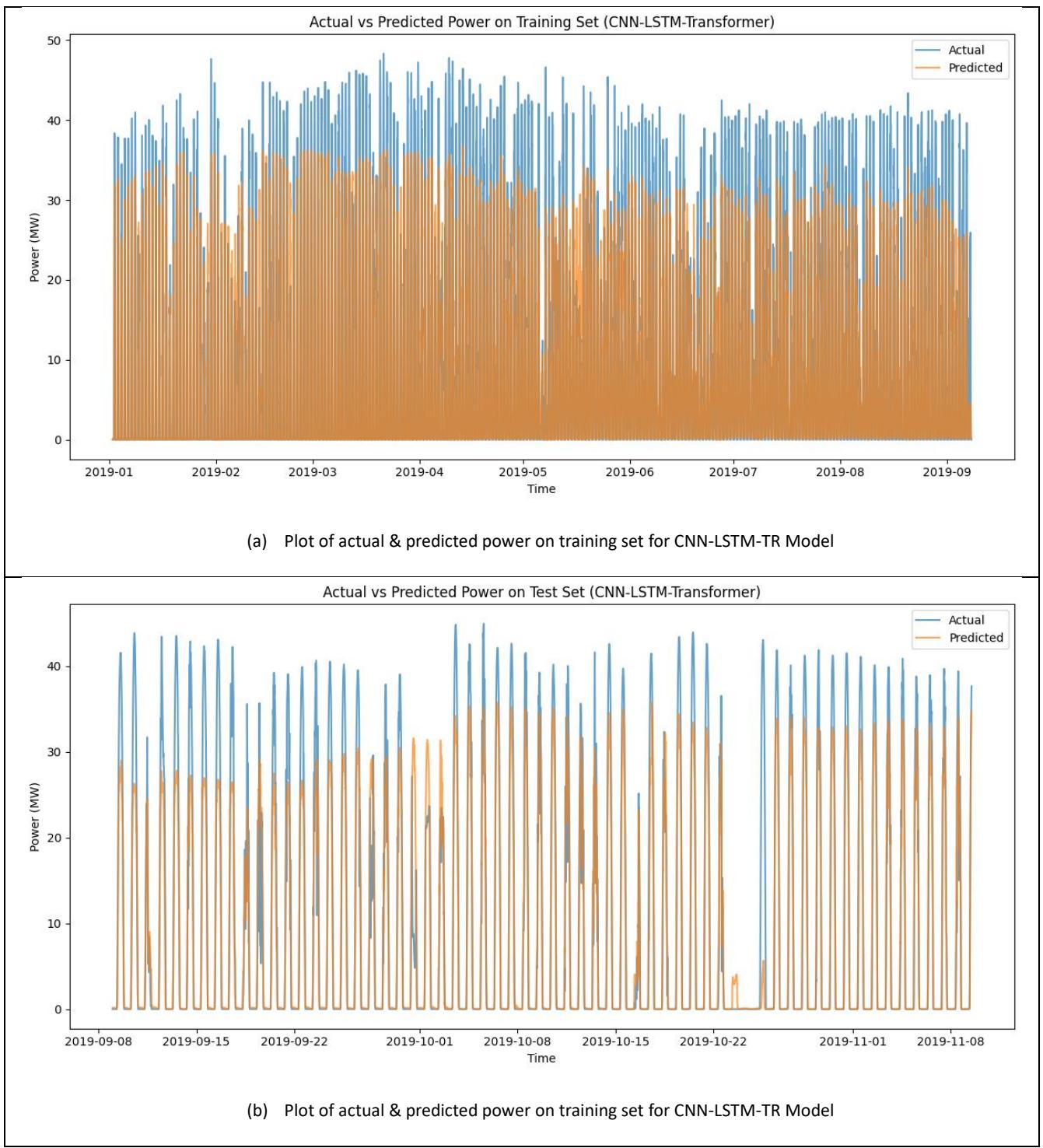
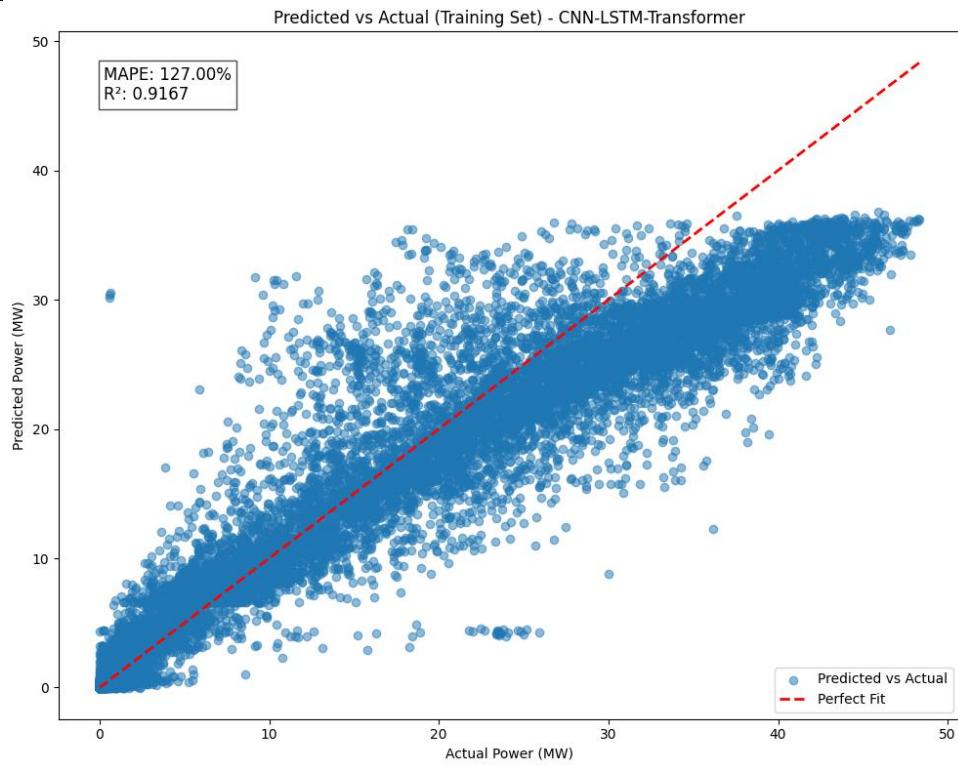
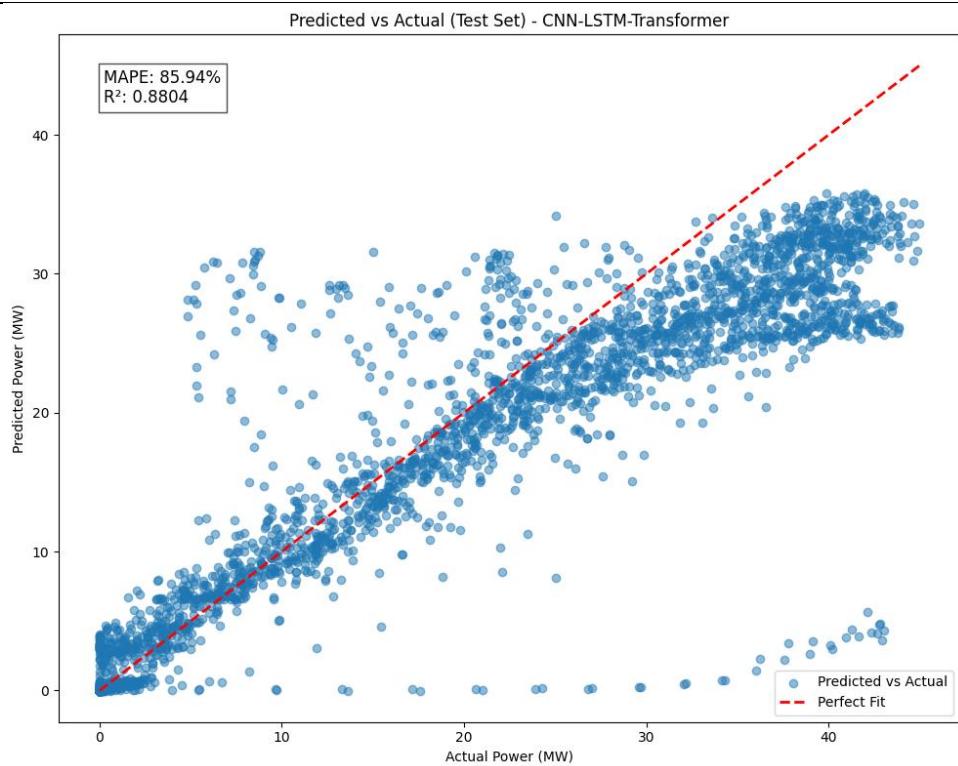


Figure A1. 21 Actual & predicted power on training & test set for CNN-LSTM-TR model



(a) Scatterplot of predicted vs actual power on training set for CNN-LSTM-TR model



(b) Scatterplot of predicted vs actual power on test set for CNN-LSTM-TR model

Figure A1. 22 Scatterplot of predicted vs actual power on training & test set for CNN-LSTM-TR model

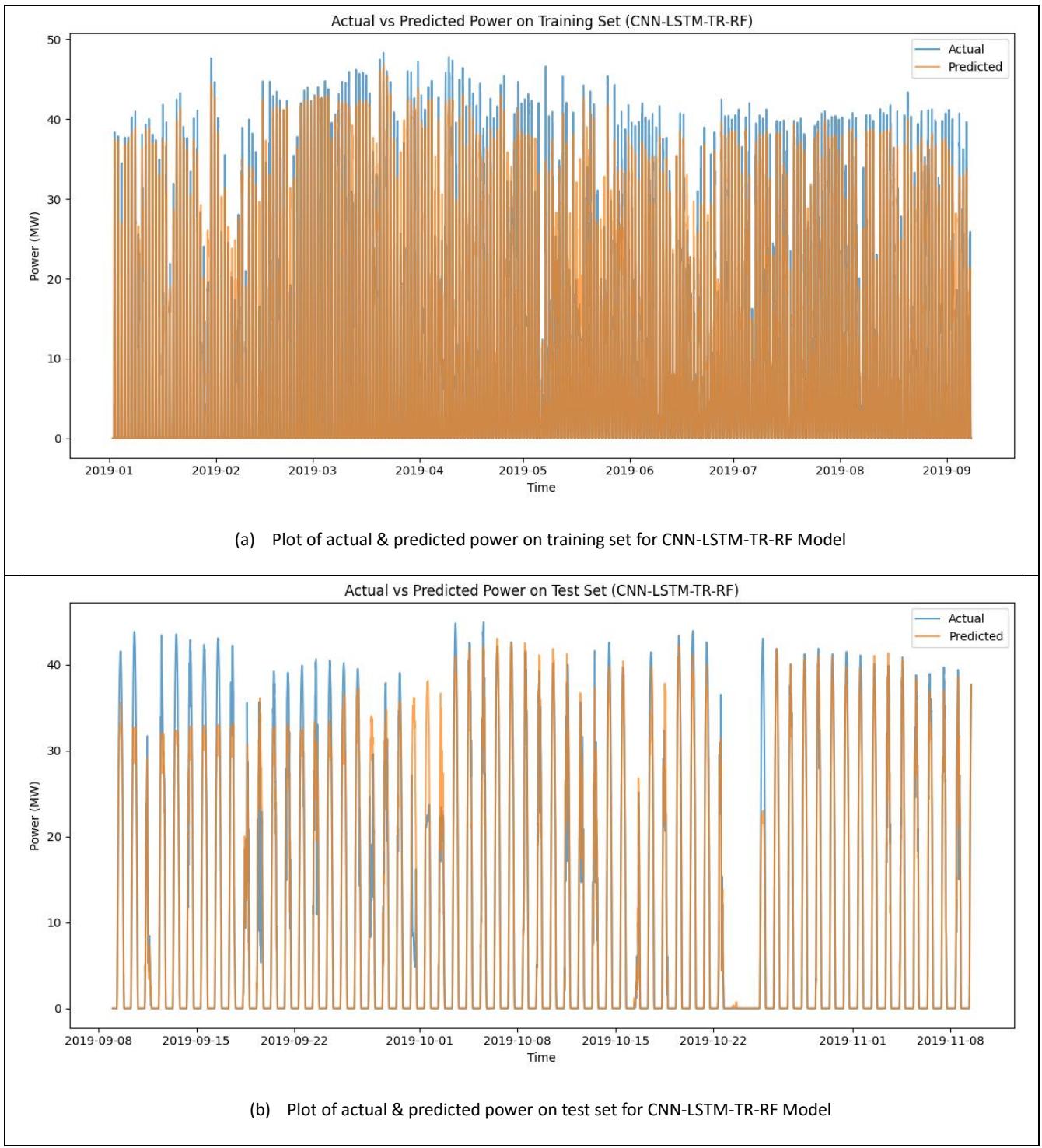


Figure A1. 23 Actual & predicted power on training & test set for CNN-LSTM-TR-RF model

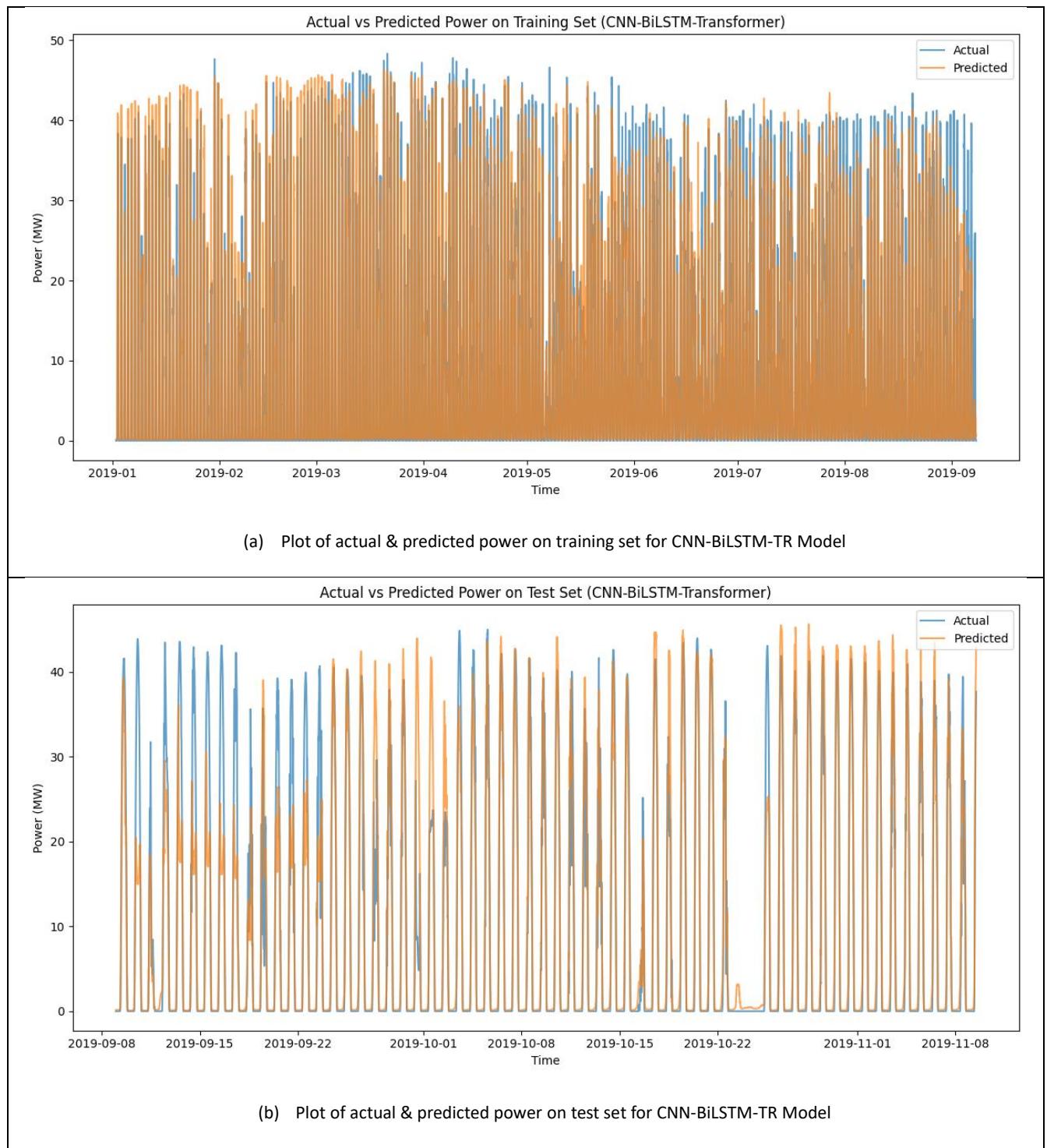
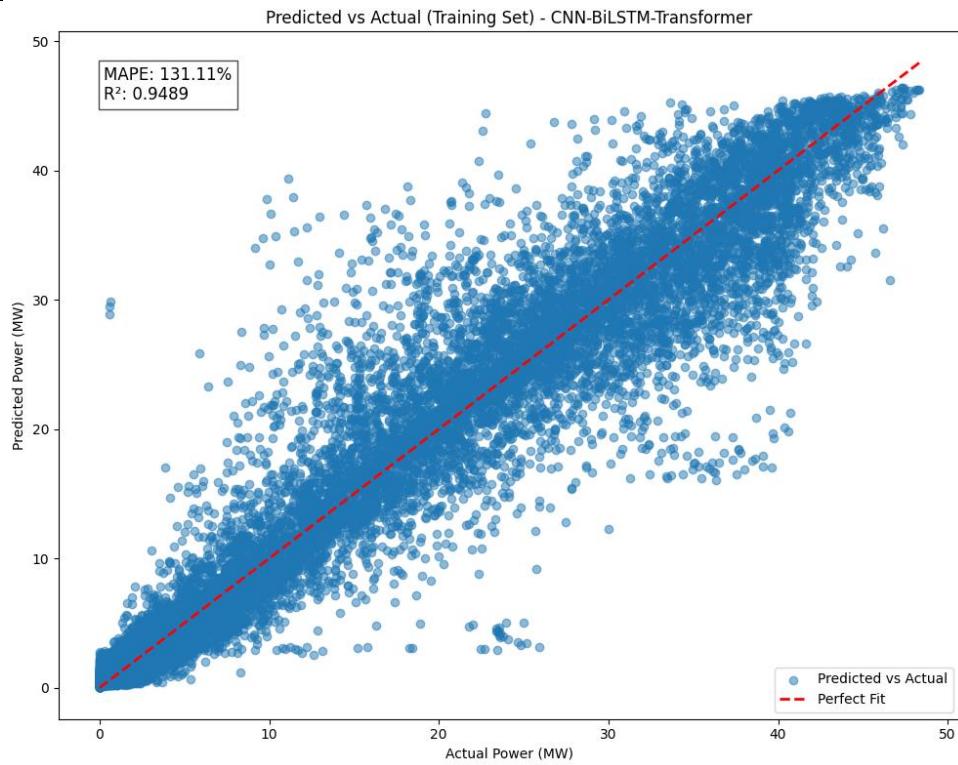
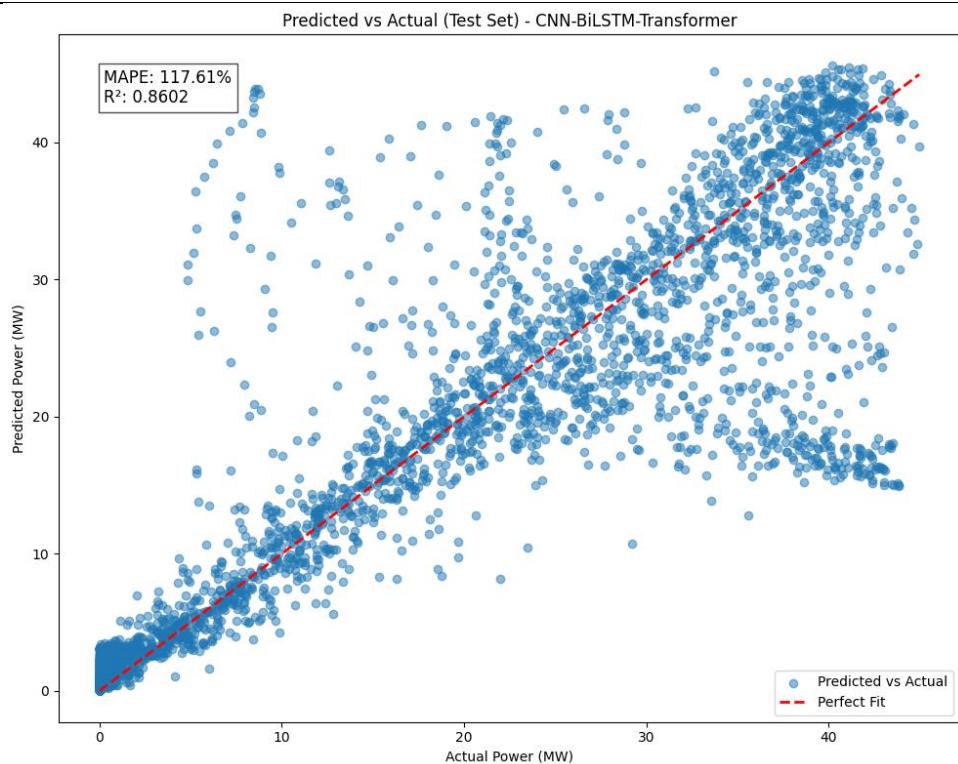


Figure A1. 24 Actual & predicted power on training & test set for CNN-BiLSTM-TR model



(a) Scatterplot of predicted vs actual power on training set for CNN-BiLSTM-TR model



(b) Scatterplot of predicted vs actual power on test set for CNN-BiLSTM-TR model

Figure A1. 25 Scatterplot of predicted vs actual power on training & test set for CNN-LSTM-TR-RF model

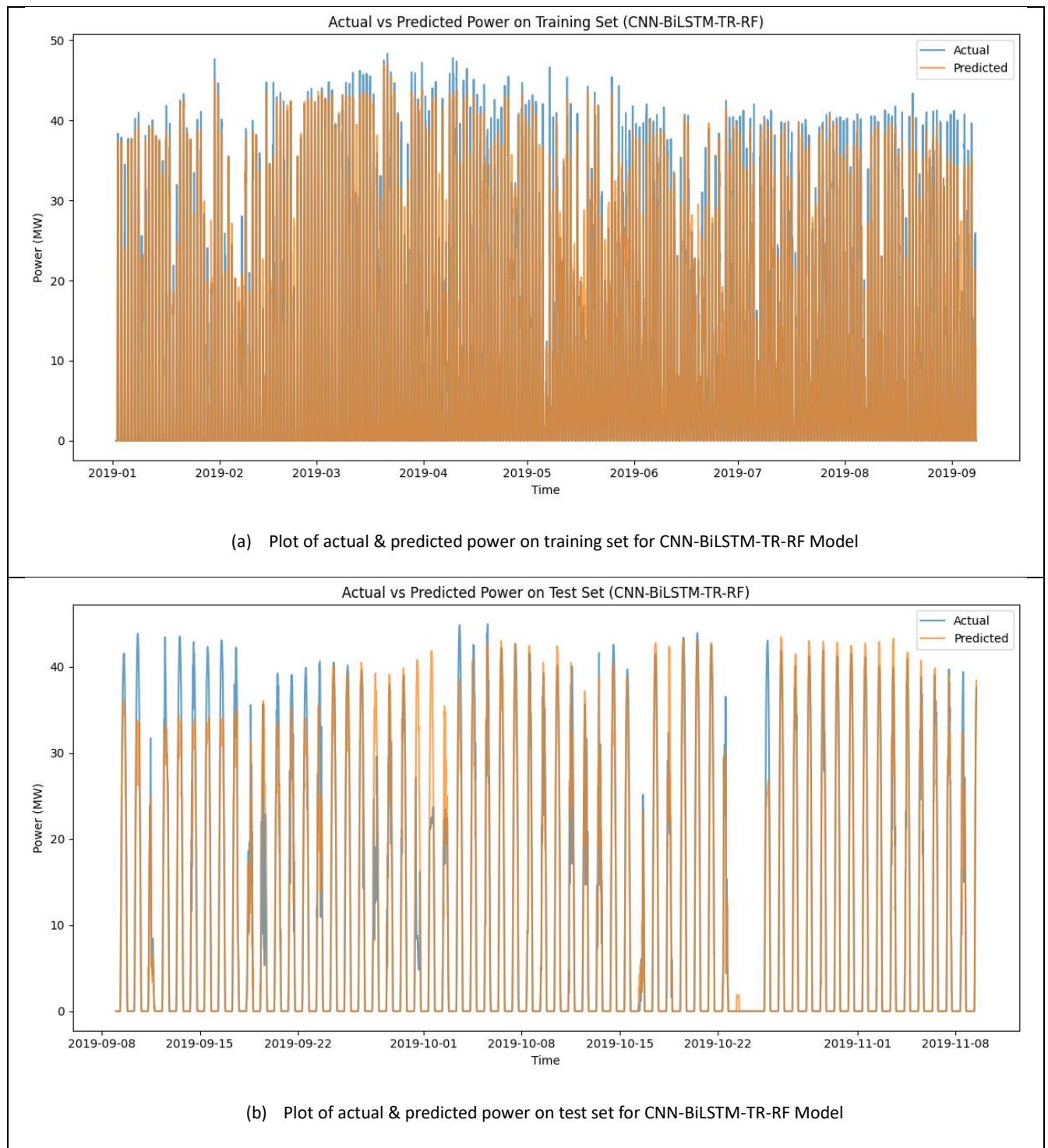


Figure A1. 26 Actual & predicted power on training & test set for CNN-BiLSTM-TR-RF model

S

## Appendix 2 Discussion

The predicted power compared to the actual power on the training and test set for the RNN model is shown in Figure A1. 1. Figure A1. 2 demonstrates strong correlation between predicted and actual values, with an R-squared value of 0.9411 and a MAPE of 94.89%, showing a slight tendency to overestimate at lower power values and underestimate at higher values (>35 MW), with dense clustering of predictions in the 20-40 MW range on training set, and the model maintains good performance on the test set with an R-squared of 0.9431 and a lower MAPE of 73.56%, exhibiting a similar pattern of slight overestimation at lower values and underestimation at higher values, with more spread in predictions across the entire range, especially noticeable in the 10-30 MW range on the test set.

The optimal hyperparameter for LSTM model are 2 LSTM layers with 32 units each, a dropout rate of 0.2, a dense layer with 32 units, learning rate of 0.001, with Adam optimizer executed over 20 epochs indicated in the model summary architecture and training & validation loss plot is shown in Figure A1. 3, the predicted power compared to the actual power on the training and test set is shown in Figure A1. 4. The scatter plot of predicted power vs actual power of the LSTM model indicates strong correlation between predicted and actual values, with an R-squared value of 0.9417 and a high MAPE of 138.80%, displaying a slight overestimation tendency at lower power values and underestimation at higher values (>40 MW) for training set, and The model maintains good correlation on the test set with an R-squared of 0.9429 and a lower MAPE of 110.53%, showing a similar pattern of slight overestimation at lower values and underestimation at higher values, with more spread in predictions, especially in the mid-range (10-30 MW), compared to the training set as shown in Figure A1. 5.s

The optimal hyperparameter for GRU model are 2 GRU layers with 64 units each, a dropout rate of 0.1, a dense layer with 16 units, learning rate of 0.0008, with RMSprop optimizer executed for 20 epochs indicated in the model summary architecture and training & validation loss plot is shown Figure A1. 6, the predicted power compared to the actual power on the training and test set is shown in Figure A1. 7. The scatter plot of predicted power vs actual power of the GRU model exhibits strong correlation between predicted and actual values, with a high R-squared value of 0.9488 and a MAPE of 122.90%, showing a slight overestimation tendency at lower power values and underestimation at higher values (>35 MW), with dense clustering of predictions in the 10-40 MW range for training set, the model maintains good correlation on the test set with an R-squared of 0.9348 and a lower MAPE of 92.46%, displaying a similar pattern of slight overestimation at lower values and underestimation at higher values, with more spread in predictions across the entire range, especially noticeable in the 20-40 MW range, and some visible outlier predictions, particularly in the lower power ranges for test set as shown in Figure A1. 8.

The optimal hyperparameter for CNN model are 1 CNN layer with filter 64 units, a kernel size of 3, a dropout rate of 0.1, 2 dense layer with 32 & 16 units respectfully, learning rate of 0.00159, with RMSprop optimizer

executed for 16 epochs indicated in the model summary architecture and training & validation loss plot is shown Figure A1. 9, the predicted power compared to the actual power on the training and test set is shown in Figure A1. 10. The scatter plot of predicted power vs actual power of the CNN model indicates moderate correlation between predicted and actual values, with a wider spread around the perfect fit line compared to previous models, as evidenced by an R-squared value of 0.9082 and a very high MAPE of 231.70%. The scatter plot reveals a clear trend of underestimation for higher power values (>30 MW) and dense clustering of predictions in the 10-30 MW range for the training set, the performance on the test set deteriorates, with the R-squared value dropping to 0.8684 and MAPE increasing to 282.18%, while maintaining the pattern of underestimation at higher power values and exhibiting more pronounced spread and outliers across the entire range, along with visible 'banding' or clustering of predictions shown in Figure A1. 11.

The scatter plot of predicted power vs actual power of the Transformer model exhibits extremely poor correlation between predicted and actual values, with a very low R-squared value of 0.0578 and an exceptionally high MAPE of 2591.21%, showing a clear inability to capture the relationship between inputs and power output, with predictions clustered mainly between 5-20 MW regardless of actual power values, and a notable vertical line of predictions around 0-5 MW actual power for the training set, the model's performance further deteriorates on the test set, with a negative R-squared value of -0.0031 and an extremely high MAPE of 1464.34%, indicating a complete failure to generalize, with predictions mostly constrained to a narrow range of approximately 5-15 MW regardless of actual power values, demonstrating no meaningful correlation between predicted and actual outputs as shown in Figure A1. 14.

The optimal hyperparameter for CNN-LSTM model are 1 CNN layer with filter 32 units, a kernel size of 5, a dropout rate of 0.1, 2 LSTM layers with 32 units each, 2 dense layer with 16 units each, learning rate of 0.000606, with RMSprop optimizer executed over 20 epochs indicated in the model summary architecture and training & validation loss plot is shown Figure A1. 15, the predicted power compared to the actual power on the training and test set is shown in Figure A1. 16. The scatter plot of predicted power vs actual power of the CNN-LSTM model demonstrates strong correlation between predicted and actual values, with a high R-squared value of 0.9442 and a MAPE of 140.74%, showing a relatively even distribution of predictions around the perfect fit line, with some underestimation tendency for higher power values (>35 MW) and slight overestimation for lower values for the training set, the model maintains good performance on the test set with an R-squared of 0.9255 and a slightly increased MAPE of 174.97%, exhibiting a similar pattern to the training set but with more spread in predictions, particularly noticeable in the mid-range (10-30 MW), and some outlier predictions visible in the lower power ranges as shown in Figure A1. 17.

With the inclusion of the best random forest model in the CNN-LSTM model, the predicted power compared to the actual power on the training and test set for CNN-LSTM-RF model is shown in Figure A1. 18. The scatter plot of predicted power vs actual power of the CNN-LSTM-RF exhibits excellent correlation between predicted and actual values, with a very high R-squared value of 0.9768 and a significantly lower MAPE of 22.37% compared to previous models, showing a tight clustering of predictions around the perfect fit line across the entire range of power values, with only minor deviations visible for the training set, the model's performance on the test set is identical to the training set, with the same R-squared value of 0.9768 and MAPE of 22.37%, indicating exceptional generalization capability and consistent prediction accuracy across both seen and unseen data, maintaining the tight clustering of predictions around the perfect fit line observed in the training set as shown in Figure A1. 19.

The optimal hyperparameter for CNN-LSTM-TR model are 2 CNN layer with filter 16 units each, 2 kernel size of 5 & 3 respectfully , a dropout rate of 0.1, 1 LSTM layers with 64 units, 2 transformer blocks with 2 attention head, 32 key dimension & 64 feed forward dimension, 2 dense layer with 64 units each, learning rate of 0.0028, with Adam optimizer executed over 12 epochs indicated in the model summary architecture and training & validation loss plot is shown Figure A1. 20, the hybrid model's predicted power compared to the actual power on the training and test set is shown in Figure A1. 21. The scatter plot of predicted power vs actual power of the CNN-LSTM-TR model demonstrates strong correlation between predicted and actual values, with a high R-squared value of 0.9167 and a MAPE of 127.00%, showing a relatively even distribution of predictions around the perfect fit line, with a slight underestimation tendency for higher power values (>35 MW) and some overestimation for lower values for the training set, the model's performance on the test set shows some degradation, with an R-squared value of 0.8804 and a lower MAPE of 85.94%, exhibiting increased spread in predictions, particularly for higher power values (>30 MW), and a more pronounced underestimation trend for these higher values compared to the training set as shown in Figure A1. 22. With the inclusion of the best random forest model in the CNN-LSTM-TR model, the predicted power compared to the actual power on the training and test set for CNN-LSTM-TR-RF model is shown in Figure A1. 23.

The CNN-BiLSTM-TR model predicted power compared to the actual power on the training and test set is shown in Figure A1. 24. The scatter plot of predicted power vs actual power of the CNN-BiLSTM-TR exhibits excellent correlation between predicted and actual values, with a very high R-squared value 0.9489 and a MAPE of 131.11%, showing a tight clustering of predictions around the perfect fit line across most of the power range, with some spread increasing for higher power values (>35 MW) for the training set, the model's performance on the test some decline, with an R-squared value of 0.8602 and a MAPE of 117.61%, indicating less accurate predictions compared to the training set. The scatter plot reveals increased spread across all power values, with notable overestimation for some mid-range values (15-30 MW) and underestimation for higher values (>35 MW), suggesting some issues with generalization to unseen data as shown in Figure A1. 25. With the inclusion of the best random forest model in the CNN-BiLSTM-TR model, the predicted power

compared to the actual power on the training and test set for CNN-BiLSTM-TR-RF model is shown in Figure A1. 26.