DEEP LEARNING FOR SOLAR POWER GENERATION FORECASTING

Project Report

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology
in
Electrical and Electronics Engineering

by

Johan Naizu (*Roll No.*: **B210787EE**)
Akash P R (*Roll No.*: **B210775EE**)
Basil K Joseph (*Roll No.*: **B210895EE**)
Afrah Rahman F (*Roll No.*: **B210929EE**)
Nikhita Krishnadas (*Roll No.*: **B210810EE**)

Under the supervision of

Dr. Nimal Madhu M

Department of Electrical Engineering

NATIONAL INSTITUTE OF TECHNOLOGY CALICUT

May 2025

CERTIFICATE

This is to certify that the report entitled "**Deep learning for Solar Power Generation Forecasting**" is a bonafide record of the **project** done by **Akash P R, Johan Naizu, Basil K Joseph, Nikhita Krishnadas** and **Afrah Rahman F** under my supervision, in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Electrical and Electronics Engineering** from **National Institute of Technology Calicut**, and this work has not been submitted elsewhere for the award of a degree.

Dr. Nimal Madhu M
Asst. Professor
Dept. of Electrical Engineering

Dr Sindhu TK
Professor & Head
Dept. of Electrical Engineering

Place : NIT Calicut
Date : 5 May 2025

ACKNOWLEDGEMENT

We would like to express my sincere gratitude to Dr. Nimal Madhu M for his invaluable support, guidance, and expertise throughout this project. We extend our appreciation to the faculty members of the Department of Electrical and Electronics Engineering for providing a conducive learning environment. We acknowledge our peers for their collaboration and constructive feedback during the preparation of this project. Thank you all for your contributions to the successful completion of this project report on Deep learning for Solar Power Generation Forecasting.

ABSTRACT

The global shift toward renewable energy has highlighted the importance of solar power generation as a sustainable energy source. However, the inherent variability of solar power due to environmental factors such as temperature, cloud cover, and solar radiation poses significant challenges for accurate forecasting. This project investigates advanced deep learning techniques to address these challenges, focusing initially on hybrid models combining Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks.

Using multivariate datasets, including weather and temporal data, the hybrid CNN-LSTM model effectively captures both spatial and temporal dependencies, delivering improved predictive accuracy. The data pre-processing pipeline includes advanced feature engineering such as lag features and rolling statistical metrics (mean, standard deviation), along with scaling and sequence formation, to ensure robust inputs to the model. Rigorous experimentation and hyperparameter optimization were conducted to enhance model performance. Evaluation metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the R^2 score demonstrated the model's ability to generalize well to unseen data.

Building upon this, the study extends into more sophisticated architectures, including Temporal Convolutional Networks (TCNs) for efficient modelling of long-term dependencies, and Transformer-based models that leverage self-attention mechanisms. To overcome limitations of attention-only models, positional embeddings were incorporated to encode temporal order, while stochastic depth was employed to improve model regularization and generalization. Finally, ensemble models were developed by integrating the outputs of CNN-LSTM, TCN, and Transformer models, yielding superior forecasting performance by mitigating individual model weaknesses.

The results underscore the potential of hybrid and ensemble deep learning frameworks in enabling accurate and reliable solar power forecasting. Such advancements are vital for effective energy management, improved grid stability, and the seamless integration of renewable energy sources into modern power systems.

CONTENTS

CHAPTER 1

INTRODUCTION

The global transition to renewable energy has highlighted the critical need for accurate solar power forecasting to maintain grid stability and optimize energy distribution. Solar power generation is inherently variable, influenced by factors such as sunlight availability, temperature fluctuations, and cloud cover. These fluctuations pose substantial challenges for precise prediction, yet accurate forecasts can significantly reduce energy storage demands, improve load scheduling, and minimize reliance on non-renewable sources

.

The dynamic and sequential nature of solar power data makes forecasting a complex task. Traditional statistical models often struggle to capture the temporal, non-linear relationships present in weather-influenced energy systems. Deep learning models have emerged as powerful tools for this purpose due to their ability to learn from complex, high-dimensional time series. This project explores the use of advanced architectures such as Long Short-Term Memory (LSTM) networks, Convolutional Neural Networks (CNNs), Temporal Convolutional Networks (TCNs), and Transformer models to model both short-term temporal dependencies and long-range contextual patterns in multivariate solar datasets. To enrich the temporal understanding of the data, we incorporated engineered features such as lag variables, rolling means, and rolling standard deviations, which capture temporal trends and local fluctuations. These features enhance the model's ability to learn patterns influenced by short-term persistence and seasonal variations

.

In addition to baseline models, the project introduces hybrid architectures, such as CNN-LSTM and Time-Distributed CNN-LSTM models, which capture both spatial and temporal features effectively. Transformer-based models were further employed with positional embeddings to retain order information and with stochastic depth to improve generalization by introducing controlled randomness in layer execution during training.

Furthermore, ensemble modelling techniques were applied to combine the predictive strengths of individual models, leading to improved robustness and generalization. Ensemble approaches such as model averaging and stacking help mitigate the limitations of single models by integrating diverse learning perspectives.

This work presents a comprehensive evaluation and comparison of these techniques through extensive experimentation, hyperparameter tuning, and performance analysis using real-world solar power and meteorological data. The findings offer practical insights into model selection and design strategies for reliable solar forecasting, contributing toward more efficient renewable energy integration.

CHAPTER 2

LITERATURE REVIEW

Accurate solar power forecasting is critical for efficient integration of renewable energy into modern power grids. However, the inherent variability caused by weather conditions and the time-dependent nature of solar irradiance make this a challenging task. Traditional statistical approaches such as ARIMA and linear regression often fall short, as they are limited in capturing the non-linear and sequential dynamics of solar power data [3][4]. These models typically assume linear relationships and fail to capture the complex interactions between multiple variables (e.g., temperature, cloud cover, solar irradiance) and their temporal evolution.

To overcome these limitations, deep learning models have emerged as powerful alternatives. Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, are effective in capturing long-term temporal dependencies and mitigating issues such as vanishing gradients [2][3]. Convolutional Neural Networks (CNNs) have also shown promise in extracting localized spatial and temporal patterns [2], making them suitable for multivariate time-series forecasting when weather variables such as temperature, cloud cover, and solar irradiance are included.

Hybrid models that combine CNNs and LSTMs leverage the strengths of both architectures. CNNs excel in capturing short-term, localized features, while LSTMs are adept at modeling long-term temporal dependencies [1][2]. However, the combination of CNNs and LSTMs is not without its challenges. The feature extraction process of CNNs can be computationally expensive, and integrating these features with LSTM's sequential processing can lead to complexity in model training and slower convergence. Despite these drawbacks, hybrid models have demonstrated substantial improvements in forecasting accuracy by capturing both local and long-range temporal dependencies. Feature engineering strategies such as cyclical encoding of time features and the inclusion of lag-based variables further enhance model performance by introducing temporal context [4][9].

Recent advancements in neural sequence modelling have introduced Temporal Convolutional Networks (TCNs), which offer a compelling alternative to RNNs. TCNs are known for their ability to capture long-range dependencies using dilated causal convolutions, offering superior training stability and parallelization benefits [7]. TCNs have the advantage of being faster to train than RNN-based models due to their non-recurrent architecture, and they can handle longer sequences more efficiently. However, their performance can be sensitive to the choice of dilation rates and the number of layers, which require careful hyperparameter tuning.

More recently, Transformer-based architectures have revolutionized time series forecasting through attention mechanisms, which allow models to focus on relevant parts of the sequence without relying on recurrence. The Transformer’s self-attention mechanism captures global temporal dependencies efficiently, especially when augmented with positional encodings that preserve the sequence order [8][12]. While Transformer models have demonstrated state-of-the-art performance, they are computationally intensive, requiring significant resources for training, especially with large datasets. The need for efficient training methods to reduce computational overhead remains a challenge for deploying Transformer models in real-time solar forecasting applications.

Enhancements such as stochastic depth have also been proposed to regularize these deep architectures and improve generalization [13]. Despite their robustness, Transformers are still prone to overfitting when dealing with noisy or highly variable input data. Introducing methods like dropout or early stopping can mitigate overfitting, but there remains a need for further regularization techniques to improve generalization in highly dynamic environments like solar power forecasting.

Another key development involves the incorporation of lag features and rolling statistics, which inject historical context and smoothed trends into the learning process. These statistical features are particularly beneficial in capturing short-term anomalies and long-term seasonality [9][10]. However, the impact of such features can vary depending on the model’s ability to capture long-term dependencies, as some architectures (e.g., CNNs) may not fully leverage these features.

Furthermore, ensemble methods have gained traction due to their robustness and ability to combine multiple model predictions to reduce variance and bias. Techniques such as Random Forests, boosting, and hybrid deep ensemble frameworks have demonstrated improved generalization across diverse solar conditions [1][5][11]. However, ensemble models can become computationally expensive, especially when integrating several deep learning models, requiring careful optimization of computational resources.

This project builds upon these state-of-the-art methods, progressing from CNN and LSTM models to TCNs and Transformers, and incorporating lag features, rolling statistics, and ensemble strategies. Through this layered approach, and by leveraging positional encodings and advanced regularization methods, the study aims to develop a robust, accurate forecasting model suited for dynamic solar power generation environments. The contributions of this work lie in combining hybrid models with ensemble strategies to mitigate the weaknesses of individual models and improve overall forecasting accuracy, demonstrating the potential of deep learning for addressing the challenges of solar power forecasting in real-world scenarios.

CHAPTER 3

METHODOLOGY

This study followed a structured, multi-phase methodology to achieve precise short-term solar power forecasting using state-of-the-art deep learning and statistical techniques. The process comprised five primary stages: data acquisition and pre-processing, feature engineering, model development, training and validation, and assessment. Each stage was meticulously designed to enhance forecasting accuracy and model robustness.

The data acquisition phase involved collecting multivariate historical data, including solar power output, temperature, cloud cover, and shortwave radiation. These variables were chosen based on their established correlation with solar irradiance patterns [4][5]. To ensure model interpretability and consistency, the data underwent pre-processing steps including normalization, missing value imputation, and time alignment [3].

Subsequently, feature engineering was applied to enrich the time-series input. This included the generation of lag features, which capture temporal dependencies by referencing prior values of power and weather parameters [9], and rolling statistics such as moving averages and rolling standard deviations to reflect smoothed trends and local variations [9][10]. Cyclical features were also derived from time components (hour, day, month) to embed seasonal and diurnal patterns [4].

The model development phase explored and compared multiple architectures:

1. LSTM networks were employed to capture long-range temporal dependencies, addressing vanishing gradients using gated memory mechanisms [2][3].
2. CNNs extracted local and spatial patterns from multivariate inputs, emphasizing short-term fluctuations [1][2].
3. A hybrid CNN-LSTM model was developed to exploit both spatial and temporal hierarchies, combining CNN pre-processing with LSTM sequential modelling [1][2].

4. Temporal Convolutional Networks (TCNs) were introduced to model sequence data using dilated causal convolutions, offering efficient training and longer memory without recurrence [7].
5. A Transformer-based model was implemented, leveraging self-attention mechanisms to globally relate input elements across time. To preserve positional information, sinusoidal positional embeddings were added to the input sequences [8][12]. Stochastic depth was incorporated during training as a regularization technique to improve generalization and stabilize deeper transformer layers [13].

To improve robustness and reduce model variance, ensemble techniques were applied. These included model averaging and weighted combinations of the outputs from LSTM, CNN-LSTM, and Transformer models [1][5][11].

In the training and validation phase, each model was trained on a fixed portion of the dataset using cross-validation. Hyperparameters such as learning rate, batch size, depth, kernel size, and attention heads were tuned through grid search and validation loss monitoring. Regularization methods, including dropout, early stopping, and weight decay, were used to mitigate overfitting and improve model generalization [4].

Model assessment was carried out using standard error metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and the coefficient of determination (R^2). Performance comparisons were conducted across all models to evaluate their predictive capabilities under consistent data conditions. The hybrid CNN-LSTM model performed reliably across varying horizons, while the Transformer-based models demonstrated strength in capturing long-term dependencies and dealing with noise in high-variance data [1][2][14].

This comprehensive methodology integrates modern time-series techniques, deep learning models, and feature engineering, forming a robust pipeline for high-resolution solar power forecasting.

CHAPTER 4

DATA COLLECTION AND PREPROCESSING

4.1 DATASET SOURCE

We used a publicly available solar power dataset, accessible via a GitHub repository. The data includes variables such as temperature (temp), cloud cover (cloud_cover), shortwave flux (shortwave_flux), and solar power output (power). The dataset was filtered to exclude records with zero power, which could represent sensor faults or missing data.

This dataset contains hourly power measurements from a 16.32 kW peak photovoltaic (PV) power plant in northern Portugal, managed by the Smart Grid and Electric Vehicle Lab (SGEVL) at INESC TEC. The system consists of three strings of 2160 W, two strings of 3360 W, and one string of 3120 W. Data spans from April 28, 2013, to June 28, 2016. Missing values (3.82%) were interpolated using measurements from other strings to ensure data completeness [6].

Weather data was sourced from MeteoGalicia's publicly available THREDDS server, which provides historical and forecasted weather variables based on the Weather Research and Forecasting (WRF) model. This data has a spatial resolution of 4 km, a temporal resolution of 1 hour, and covers up to 96 hours in advance. The geographical area spans 2400 km^2, represented by a 13 × 13 grid of 169 evenly distributed points [6].

Link to dataset used: https://rdm.inesctec.pt/dataset/pe-2020-002

4.2 DATA FILTERING

The dataset underwent a rigorous preprocessing pipeline to remove irrelevant or noisy data points to ensure the extraction of meaningful and relevant features for the solar power generation forecasting task. The first step involved standardizing the timestamp column into a consistent datetime format, which was then set as the index to facilitate temporal manipulation and ensure proper alignment of the time series data. To capture seasonal patterns in the data, time-based features like Day sin, Day cos, Year sin, and Year cos were engineered using trigonometric functions [5][6]. These features help encode cyclical variations in daily and annual patterns, essential for accurately modeling solar power generation.

Initially, zero-output data points—where the recorded solar power was zero—were excluded from the dataset under the assumption that these values primarily corresponded to periods of no solar irradiation, such as nighttime or dense cloud cover. The rationale was to eliminate non-informative samples and focus training on periods of active power generation. However, empirical results showed that this exclusion led to a noticeable degradation in model accuracy. The absence of zero output instances disrupted the model's ability to learn the natural diurnal cycles and periods of inactivity that are inherent to solar power systems. Consequently, this step was reversed, and zero-value entries were retained in the dataset. Including these points provided the model with a more comprehensive understanding of real-world solar generation behavior, including transitions between active and inactive states, which ultimately improved forecasting accuracy and robustness.

4.3 FEATURE SELECTION

Following data filtering, feature selection was conducted based on the known physical relationships between the selected variables and solar power generation. The chosen features were:

4.3.1 Temperature: Temperature directly influences the efficiency of solar panels. Higher temperatures can lead to a reduction in panel efficiency, affecting power generation output.

4.3.2 Cloud Cover: The extent of cloud cover influences the amount of sunlight reaching the solar panels. Increased cloud cover leads to a reduction in solar irradiance, thereby lowering the power output.

4.3.3 Shortwave Flux: Shortwave flux is a direct measure of solar radiation. This feature is crucial as it quantifies the amount of solar energy available to the panels, which in turn impacts the power generated.

4.3.4 Power Output: The target variable for prediction, which represents the actual solar power output. To facilitate forecasting, the power output feature was shifted by one timestep, enabling the model to predict the subsequent timestep's power output based on historical data.

4.4 TEMPORAL PATTERN ANALYSIS

To assess the temporal dynamics and interdependencies of the selected features, a detailed pattern analysis was performed. This involved visualizing the features over time to identify significant trends, cycles, and correlations. Graphs were illustrated to study the temporal variations in the features.

4.4.1 Temperature Time Series

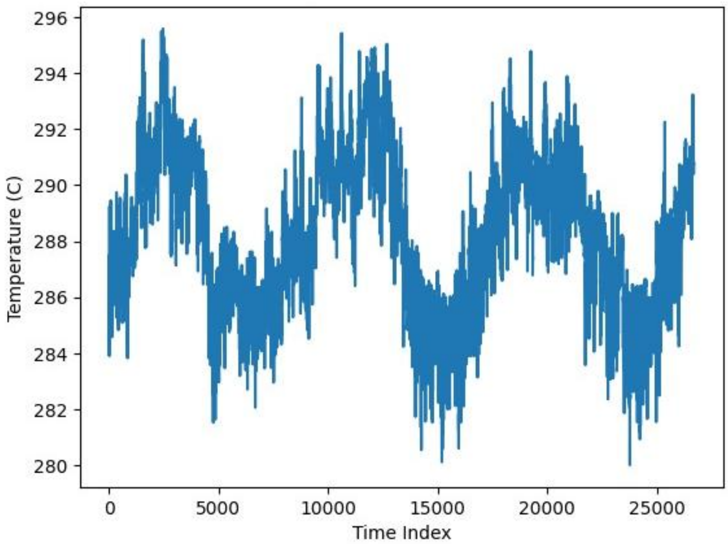

Figure 4.1: Temperature Time Series

The temperature exhibits a clear daily cycle, with higher temperatures typically observed during the day and lower temperatures at night. This pattern reflects the natural heating and cooling of the environment due to the Earth's rotation and solar radiation during daylight hours. The fluctuations in temperature have a direct relationship with solar power output. Higher temperatures, although indicative of better solar radiation during the day, can also reduce the efficiency of solar panels. Therefore, understanding these temperature variations is vital for predicting the energy output accurately.

4.4.2 Cloud Cover Time Series

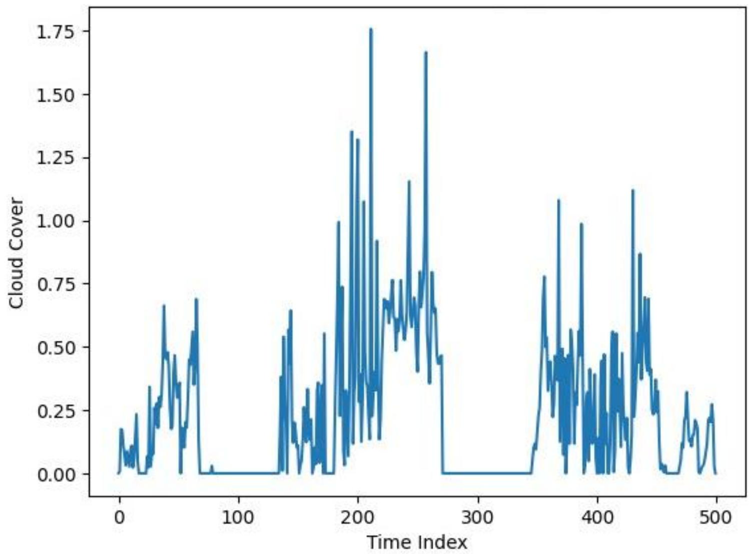

Figure 4.2: Magnified Cloud Cover Time Series

The cloud cover exhibits a pronounced diurnal variation, with higher cloud density typically observed during the late afternoon and evening hours, and reduced cloud cover during the early morning and midday. This pattern is indicative of the atmospheric conditions driven by local weather systems, such as the daily heating and cooling cycles that cause convection, leading to cloud formation in the afternoon. These fluctuations in cloud cover are closely tied to the dynamics of atmospheric moisture, temperature, and wind patterns, and they significantly impact solar irradiance.

During periods of higher cloud cover, solar power generation is likely to be reduced due to the diminished direct sunlight reaching the solar panels. The intensity of this reduction varies depending on cloud type and thickness, with thicker clouds leading to more substantial decreases in power output. Conversely, during times of reduced cloud cover, particularly midday when solar radiation is at its peak, solar power generation tends to increase, following a clearer path of sunlight.

4.4.3 Shortwave Flux Time Series

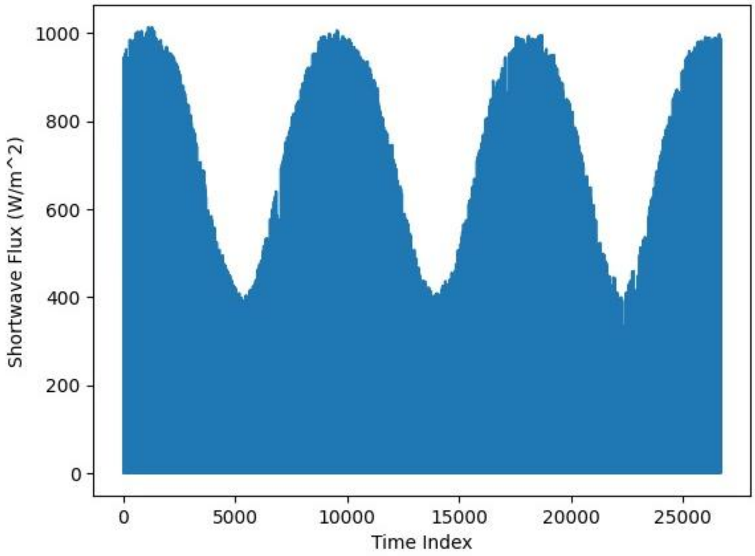

Figure 4.3: Shortwave Flux Time Series

The shortwave flux, which represents the incoming solar energy, follows this daily variation, with a gradual increase from sunrise, reaching its maximum around noon, and then decreasing as the sun sets. This cyclical fluctuation in shortwave flux is closely tied to the Earth's axial tilt and orbital dynamics, dictating the intensity and duration of solar radiation at any given location throughout the day. The maximum shortwave flux corresponds to the period of highest solar insolation, which is critical for solar power generation.

Variations in the shortwave flux time series can also be influenced by other environmental factors, such as cloud cover and atmospheric scattering, which can attenuate the incoming solar radiation. Consequently, fluctuations in shortwave flux are directly related to changes in cloud cover, with lower flux values typically associated with overcast or cloudy conditions.

4.4.4 Power Output Time Series

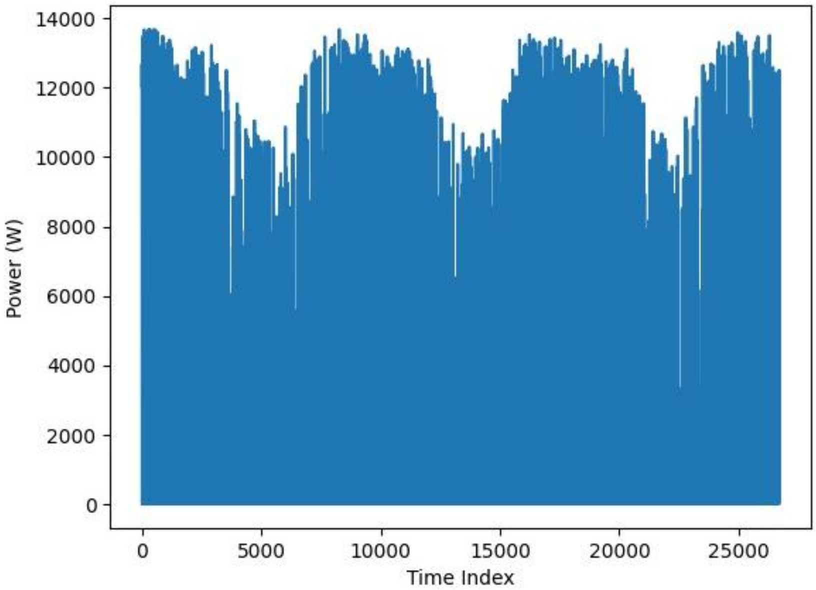

Figure 4.3: Power Output Time Series

The power output exhibits a clear diurnal variation, closely mirroring the fluctuations in shortwave flux and cloud cover. Peak power output occurs during midday when solar radiation is at its highest, reflecting optimal conditions for solar panel energy generation. Conversely, the power output tends to be lower during the early morning and late evening hours, when the sun's position is at an angle, and the solar irradiance is significantly reduced.

The time series also reveals that power output is not only influenced by the solar radiation levels but also by the efficiency of the solar panels, which can be affected by factors such as temperature, dust, or panel degradation. The shift in power output by one timestep, as used for prediction purposes, highlights the system's inherent temporal dependencies, where past power output values are predictive of future values. These dependencies are particularly important in modeling and forecasting scenarios, as the power output at any given time is often influenced by the output from preceding hours.

4.5 DATA SCALING AND NORMALIZATION

To ensure compatibility with deep learning models, which are sensitive to the range of input data, all features and the target variable were scaled between 0 and 1 using MinMaxScaler. This scaling ensures that all features contribute proportionally during training and prevents any one feature from dominating the learning process due to differences in scale [2][5].

4.6 CONVERTING 2D TO 3D FOR TIME-SERIES INPUT

The dataset was restructured into a 3D format to capture temporal dependencies, essential for time-series forecasting. Each sample comprises sequential feature data from a fixed number of previous time steps, forming sliding windows of input data [1][5]. This restructuring pairs these sequences with corresponding target values from the subsequent time step, enabling models to learn patterns across time effectively.

4.7 TRAIN-TEST SPLIT AND SEQUENCE FORMATION

To prepare the dataset for training and evaluation, an 80-20 split was applied to divide the data into training and testing subsets. This ensures the model is trained on a substantial portion of the data while leaving enough unseen data for performance evaluation [2][3]. Additionally, the time-series data was structured into sequences with a 24-hour time step. This allows the model to capture temporal dependencies effectively, enabling accurate predictions based on past patterns and trends [4]. This sequence formation is crucial for leveraging the time-series nature of the data to enhance forecasting accuracy

CHAPTER 5

MODEL IMPLEMENTATION

5.1 MODEL COMPARISONS

The project began with a comparative evaluation of several deep learning architectures tailored for multivariate time-series solar power forecasting. The initial models included Recurrent Neural Networks (RNNs), which demonstrated a foundational ability to capture temporal dependencies but struggled with vanishing gradients and limited memory retention [1][3]. Long Short-Term Memory Networks (LSTMs) significantly improved upon this by incorporating gating mechanisms to preserve long-term temporal information, offering superior results for time-dependent solar features [2][4]. Simultaneously, Convolutional Neural Networks (CNNs) were used to extract short-term and localized patterns from weather and irradiance data, proving effective in identifying critical spatial-temporal features [1][3].

Beyond these, a Temporal Convolutional Network (TCN) was introduced to further enhance sequence modelling. TCNs, with their dilated causal convolutions and residual connections, enabled the model to capture both short- and long-term dependencies efficiently, all while benefiting from parallel computation and stable gradient flow [7][10][13].

Most recently, the project explored a Transformer-based architecture, which replaced recurrence entirely with self-attention mechanisms. The model leveraged positional embeddings to encode temporal order and stochastic depth for improved regularization during training. This approach offered strong performance in long-horizon forecasting tasks and proved particularly adept at learning global dependencies across time without being constrained by fixed receptive fields, outperforming previous architectures in some scenarios [11][12][14].

5.1.1 Recurrent Neural Networks (RNNs)

RNNs process sequential data by maintaining a hidden state that updates as the input sequence progresses. However, the challenges of vanishing and exploding gradients limited their application in handling longer sequences, leading to reduced forecasting accuracy[1][2].

5.1.2 Long Short-Term Memory Networks (LSTMs)

LSTMs, an advanced variant of RNNs, address these challenges by incorporating gating mechanisms, such as forget, input, and output gates. This allows LSTMs to selectively retain and discard information, resulting in better modelling of long-term dependencies and higher accuracy in temporal data forecasting[2][3][4].

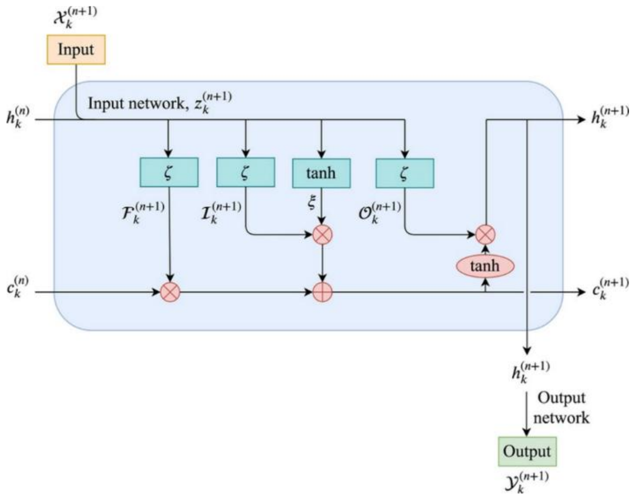

Figure 5.1: LSTM Architecture

5.1.3 Convolutional Neural Networks (CNNs)

Traditionally used for image data, CNNs proved valuable in this project for analysing spatial relationships among features. By applying convolutional filters to extract temporal patterns, CNNs contributed complementary strengths to forecasting tasks [1][5].

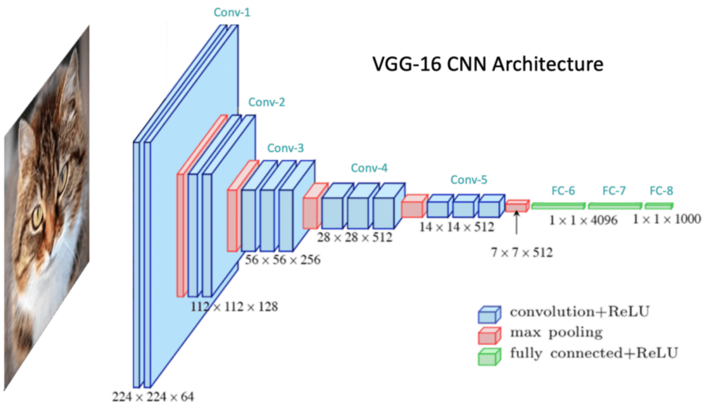

Figure 5.2: CNN Architecture

5.1.4 Temporal Convolutional Networks (TCNs)

TCNs utilize stacked causal and dilated convolutions to model temporal dependencies in sequence data. Their non-recurrent architecture enables stable gradients, faster training, and a flexible receptive field, making them well-suited for time-series forecasting. In this project, TCNs effectively captured both short-term and long-range temporal patterns in solar power data [7][10].

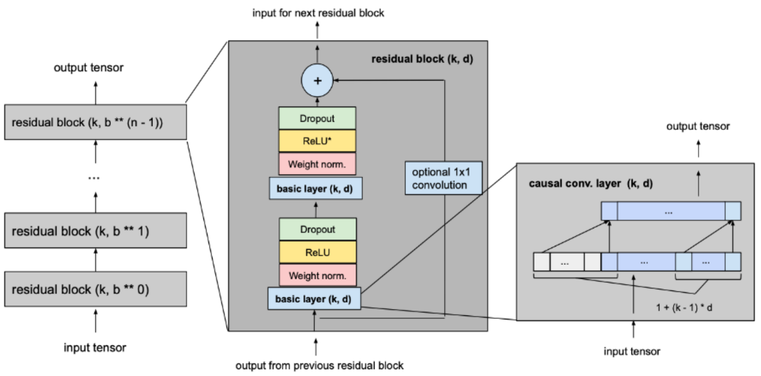

Figure 5.3: TCN Architecture

5.1.5 Transformer Architecture

Originally designed for natural language processing, the Transformer architecture has shown strong performance in time-series forecasting by leveraging self-attention mechanisms. It captures global temporal dependencies efficiently without relying on recurrence. In this project, Transformers enabled the model to attend to relevant time steps dynamically, improving accuracy in forecasting highly variable solar power generation [8][9][11].

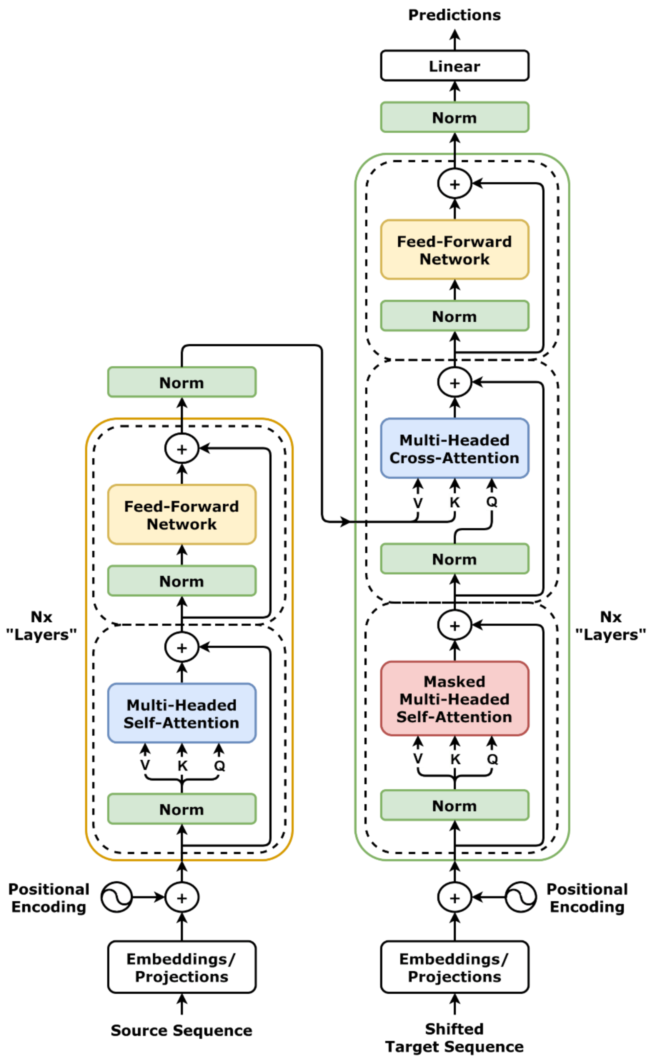

Figure 5.4: Transformer Architecture

5.2 IMPLEMENTATION OF LSTM

The Long Short-Term Memory (LSTM) model is designed to capture temporal dependencies in sequential data, making it well-suited for forecasting tasks like solar power generation. Unlike traditional feedforward networks, LSTMs are capable of retaining important information over long sequences, mitigating issues like vanishing gradients, and modelling time-dependent patterns in the data. The LSTM model consists of several layers designed to process and predict time-series data. The following table summarizes the architecture of the LSTM model.

Layer	**Output Shape**	**Parameters**
Input Layer	(None, *time_steps*, *n_features*)	0
LSTM	(None, 50)	20,400
Dropout	(None, 50)	0
Dense	(None, 50)	2,550
Dense	(None, 1)	51

Table 5.1: LSTM Model Architecture

This LSTM-based model processes the temporal data sequentially and makes predictions based on learned dependencies. It is particularly effective for solar power forecasting tasks, where the output is dependent on past time steps. The architecture combines the advantages of the LSTM's ability to model sequential dependencies with dropout regularization to enhance generalization performance [2][4].

5.3 IMPLEMENTATION OF CNN

The Convolutional Neural Network (CNN) model is designed to extract spatial features from the input time-series data by applying convolutional filters. CNNs are particularly effective in detecting local patterns in data, making them ideal for the initial feature extraction step in the hybrid model for solar power forecasting. The model employs multiple convolutional layers followed by pooling, regularization, and fully connected layers to make accurate predictions based on these spatial features. The CNN model consists of several layers, starting with convolutional layers to extract features and followed by pooling, dropout, and dense layers for further processing.

Layer	Output Shape	Parameters
Input Layer	(None, time_steps, n_features)	0
Conv1D	(None, time_steps-2, 128)	4,832
Conv1D	(None, time_steps-4, 64)	24,832
MaxPooling1D	(None, (time_steps-4)/2, 64)	0
Dropout	(None, (time_steps-4)/2, 64)	0
Flatten	(None, 64*(time_steps-4)/2)	0
Dense	(None, 64)	4,160
Dropout	(None, 64)	0
Dense	(None, 16)	1,040
Dropout	(None, 16)	0
Dense	(None, 1)	17

Table 5.2: CNN Model Architecture

By applying multiple convolutional filters, the model can efficiently learn the local temporal patterns that are essential for accurate solar power forecasting. The combination of convolution, pooling, dropout, and dense layers enhances the model's ability to capture complex relationships in the data while minimizing overfitting [1][5].

5.4 HYBRID MODEL DEVELOPMENT

The hybrid model combines the strengths of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks to enhance solar power forecasting accuracy. The development process involves two key stages:

5.4.1 Feature Extraction with CNN: CNN layers are utilized initially to extract meaningful spatial patterns from the multivariate time-series data. These patterns, representing local temporal dependencies and relationships among variables, are captured by applying convolutional filters over the input sequences[1][5]. This step reduces noise and emphasizes relevant features, creating an efficient representation of the input data.

5.4.2 Sequential Modelling with LSTM: The feature maps generated by the CNN are passed to LSTM layers. LSTMs, designed to handle sequential dependencies, process the temporal structure in the extracted features. By preserving information over long time horizons and addressing challenges like vanishing gradients, LSTMs refine the temporal understanding of the data, enabling better predictions[2][4].

Layer	**Output Shape**	**Parameter**
CNN	(None, 3, 6, 64)	1600
Flatten	(None, 3,384)	0
LSTM	(None, 128)	262656
Dropout	(None,128)	0
Dense	(None,1)	129

Table 5.3: Hybrid Model Architecture

This hybrid approach leverages the CNN's ability to distil raw data into compact representations and the LSTM's capability to model time-dependent relationships. By combining these architectures, the model achieves a balance between spatial pattern recognition and temporal sequence learning, improving forecasting performance. The hybrid architecture is particularly well-suited for solar power prediction, where input variables exhibit both spatial and temporal interdependencies [1][3][5].

5.5 TIME DISTRIBUTED CNN-LSTM HYBRID MODEL

Before feeding the data into the model, the input sequences are reshaped into subsequences. This reshaping divides the data into smaller chunks, allowing the model to process them in segments while capturing both spatial and temporal patterns. After reshaping the data, the model is trained and evaluated. The input shape for the model is (subsequences, inner_timesteps, n_features), where each subsequence is processed independently by the convolutional layers, followed by sequential processing by the LSTM layers.

This architecture captures both spatial features using convolutional layers and temporal dependencies using LSTM layers. The TimeDistributed wrapper ensures that the convolutional operations are applied to each subsequence, while the LSTM layers aggregate these local patterns over time to make accurate predictions for solar power generation. This hybrid approach effectively combines the strengths of both CNNs and LSTMs, making it a robust choice for time-series forecasting tasks [1][2][3][5].

Layer	Output Shape	Parameters
Input Layer	(None, subsequences, inner_timesteps, n_features)	0
TimeDistributed Conv1D	(None, subsequences, inner_timesteps-2, 32)	3,200
TimeDistributed Conv1D	(None, subsequences, inner_timesteps-4, 64)	6,144
TimeDistributed Conv1D	(None, subsequences, inner_timesteps-6, 128)	24,832
TimeDistributed Flatten	(None, subsequences, (inner_timesteps-6)*128)	0
LSTM	(None, subsequences, 128)	131,584
LSTM	(None, 64)	49,152
Dropout	(None, 64)	0
Dense	(None, 64)	4,160
Dropout	(None, 64)	0
Dense	(None, 1)	65

Table 5.4: Time-Distributed CNN-LSTM Hybrid Model Architecture

5.6 TEMPORAL CONVOLUTIONAL NETWORK (TCN)

The Temporal Convolutional Network (TCN) is a deep learning architecture designed specifically for sequence modeling, particularly effective in time series forecasting tasks. Unlike recurrent networks, TCNs rely entirely on convolutional layers, allowing them to model temporal dependencies while offering faster training due to parallelization. The core components of a TCN include causal convolutions, which ensure that each time step prediction is only influenced by previous time steps, preserving the temporal order; dilated convolutions, which exponentially increase the receptive field without adding extra parameters; and residual connections, which improve gradient flow, allowing for deeper networks without degradation of performance [7][13].

5.6.1 Temporal Dependency Modeling with Dilated Causal Convolutions: In the TCN model, dilated causal convolutional layers are applied to the input sequences. These layers progressively expand the receptive field, capturing both short-term fluctuations and long-range dependencies. By stacking multiple layers with increasing dilation factors (such as 1, 2, 4, and 8), the network aggregates information from a broad temporal context, essential for accurate solar power forecasting, where past weather data and irradiance levels strongly influence future solar power generation [7][13]. This dilation technique allows the model to learn from diverse time scales without significantly increasing the computational burden.

5.6.2 Feature Compression and Prediction: After processing the input with convolutional layers, the output feature maps are directly passed through fully connected Dense layers (with 128 neurons in this case). This step transforms the extracted temporal features into a fixed-length representation. The final output layer, with a single neuron, produces the forecasted power output. This design ensures that the model remains computationally efficient while capturing complex temporal relationships within the data [10].

Layer	Output Shape	Parameters
Input Layer	(None, time_steps, n_features)	0
TCN (dilations=[1, 2, 4, 8], nb_filters=64)	(None, 64)	88,640
Dense	(None, 128)	8,320
Dropout	(None, 128)	0
Dense	(None, 1)	129

Table 5.5: TCN Model Architecture

This TCN-based model applies temporal convolutional layers with increasing dilation factors, enabling the network to capture wide-ranging temporal dependencies. The feature maps generated by the TCN are then passed through fully connected dense layers, which refine the representation and ultimately predict the solar power output. The TCN model has demonstrated promising results in modeling complex temporal relationships in solar power forecasting, offering an efficient and scalable alternative to LSTM-based architectures [3][5].

5.7 TRANSFORMER-BASED MODEL DEVELOPMENT

Transformer models, originally introduced in the context of natural language processing (NLP), have revolutionized the field of sequence modelling by leveraging attention mechanisms to capture long-range dependencies. Unlike recurrent models such as RNNs and LSTMs, which process sequences sequentially, Transformers allow for parallelized computations by simultaneously attending to all time steps. This enables them to efficiently learn global dependencies, making them highly suitable for time-series forecasting tasks, including solar power generation prediction, where dependencies can span long temporal horizons, including daily and seasonal patterns.

In this project, the Transformer architecture is adapted for solar power forecasting, utilizing the following key features:

5.7.1 Self-attention mechanisms: The Transformer relies on self-attention, a mechanism that computes a set of attention weights dynamically for each time step in the sequence. By attending to the entire sequence simultaneously, the model can effectively focus on relevant time steps, regardless of their position in the sequence. This is particularly useful for modeling complex temporal relationships in solar power data, where past weather conditions and irradiance values influence future power generation over long periods. The self-attention mechanism enables the model to capture global relationships without the constraints of recurrence [8][12].

5.7.2 Positional encoding: Since the Transformer architecture does not inherently model the sequential order of input data, positional embeddings are added to preserve the temporal structure of the sequence. These embeddings encode the relative or absolute position of time steps, allowing the model to differentiate between earlier and later points in the sequence. In this implementation, learnable positional embeddings are applied to the input data, enhancing the model's ability to handle sequential information in time-series forecasting tasks [7][12]. The input projection step ensures that the embedding dimension aligns with the number of features in the input data, facilitating efficient learning.

5.7.3 Transformer Blocks with DropPath Regularization: The architecture consists of stacked Transformer blocks, each containing a multi-head self-attention layer followed by a feed-forward neural network (FFN). To prevent overfitting and improve generalization, DropPath regularization is applied within each Transformer block. This technique randomly drops entire paths during training, which encourages the model to learn more robust features and reduces the risk of overfitting, particularly in deep models [13]. This addition significantly enhances the model's ability to generalize to unseen data, a crucial aspect in forecasting tasks like solar power prediction, which often involves noisy and highly variable input data.

5.7.4 Scalability and Parallelization: One of the key advantages of the Transformer model is its ability to process sequences in parallel, unlike recurrent models that process data sequentially. This parallelization not only speeds up training but also improves computational efficiency, especially for large-scale datasets. Given the temporal nature of solar power data, which often includes a wide range of time steps (e.g., hourly, daily, seasonal), the Transformer model's scalability is critical for handling such high-dimensional data efficiently [8][11].

Layer	Output Shape	Parameters
Input Layer	(None, time_steps, n_features)	0
Positional Embedding	(None, 12, 20)	240
Transformer Block	(None, 12, 20)	9,384
Transformer Block	(None, 12, 20)	9,384
Layer Normalization	(None, 12, 20)	40
GlobalAveragePooling1D	(None, 20)	0
Dense	(None, 64)	3,250
Dropout	(None, 64)	0

Dense	(None, 32)	2,080
Dropout	(None, 32)	0
Dense	(None, 1)	33

Table 5.6: Transformer-Based Model Architecture

This model leverages multi-head attention to capture complex temporal dependencies across the input sequence. The attention mechanism allows the model to dynamically focus on relevant time steps, regardless of their position in the sequence, enabling it to effectively model both short-term fluctuations and long-term patterns in solar power generation data. The inclusion of feedforward layers after the attention mechanism, combined with residual connections, ensures stable training and promotes gradient flow, which is essential for deeper architectures. These components together help improve the model's generalization by allowing it to learn robust features from diverse input sequences, mitigating issues like overfitting [7][12].

Positional encodings are integrated into the input data to preserve temporal structure, as Transformers lack a built-in notion of sequence order. This feature is essential for forecasting tasks, where understanding the relative timing of events (such as solar irradiance or temperature changes) is crucial for accurate predictions [12][13].

In practice, the Transformer-based model demonstrated superior performance in capturing long-range dependencies and sudden variations in solar power output. These capabilities make it an ideal choice for tasks that involve highly variable time-series data, such as solar power forecasting. The model's ability to process the entire sequence in parallel, along with its attention mechanism, positions it as a competitive alternative to traditional sequential models like LSTMs, offering both efficiency and accuracy in handling complex temporal dynamics [4][5][8].

5.8 FEATURE ENGINEERING

To enhance model performance and provide richer input representations, a comprehensive feature engineering pipeline was implemented. The focus was on capturing temporal cycles, historical dependencies, and recent trends relevant to solar power generation. Temporal encodings were introduced using sine and cosine transformations to represent cyclical patterns of time, allowing the model to better learn the periodic nature of solar irradiance. These included daily (Day sin, Day cos), yearly (Year sin, Year cos), hourly (hour_sin, hour_cos), and monthly (month_sin, month_cos) components [4][5][7].

Lag features were created to provide historical context by referencing previous time steps. These features help the model understand how recent weather and power generation patterns influence future output. Lags were applied to power, temperature, cloud cover, and shortwave flux variables. Rolling window features were added to summarize recent fluctuations in power output. These included the rolling mean and standard deviation over a short window, which enabled the model to learn local trends and variability [3][6][8].

Feature Type	**Features**
Time Encodings	Day sin, Day cos, Year sin, Year cos, hour_sin, hour_cos, month_sin, month_cos
Lag Features	power_lag_1, power_lag_2, power_lag_3, power_lag_6, temp_lag_1, cloud_cover_lag_1, shortwave_flux_lag_1
Rolling Statistics	power_roll_mean_3, power_roll_std_3
Core Meteorological	temp, cloud_cover, shortwave_flux

Table 5.7: Engineered Features

While these engineered features were designed to capture a range of temporal and meteorological patterns, they did not result in significant improvements in model performance. The addition of lag and rolling window features, as well as the temporal encodings, provided some value but did not lead to large changes in the overall accuracy or robustness of the models. This suggests that while the engineered features contributed additional context, the base model architecture was already capturing most of the relevant patterns. Nevertheless, these features may still provide subtle benefits and could be further optimized in future iterations, ensuring that the model remains adaptable to a broader range of scenarios [12][14].

5.9 ENSEMBLE MODELING APPROACH

To further enhance the robustness and forecasting accuracy of solar power generation, an ensemble modeling approach was employed. Ensemble methods combine the predictions of multiple base models to mitigate individual model weaknesses, exploit diverse learning patterns, and improve generalization. In this study, the ensemble approach was designed to integrate the complementary strengths of two distinct deep learning architectures—the Temporal Convolutional Network (TCN) and Transformer models, each of which excels in different aspects of time-series forecasting [3][4].

5.9.1 Motivation for Ensembling

While individual models such as TCNs are well-suited for capturing local temporal dependencies and short-term fluctuations in time-series data, Transformers excel in modeling long-range dependencies, especially in the presence of complex, non-linear patterns. However, no single model was found to consistently outperform in all scenarios. The Transformer's reliance on an attention mechanism offers superior performance for long-sequence modeling but is highly sensitive to noise and prone to overfitting under certain conditions [8][12]. Conversely, TCNs provide robust performance on shorter sequences but may struggle with long-term forecasting due to their limited receptive field [14]. Therefore, combining these two models allows the ensemble to leverage their complementary strengths while compensating for their individual weaknesses, resulting in a more stable and accurate forecasting model [7][11].

5.9.2 Ensemble Strategy

The ensemble methodology adopted in this study involved the integration of the TCN and Transformer models via a regression-based aggregation technique. Predictions from the two base models were fed into a regression model, which learned to optimally combine the individual model outputs. The regression model was trained on the validation set using error metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) to determine the weight of each model's contribution to the final prediction. This strategy allowed the ensemble to capitalize on the unique strengths of both models, while minimizing the impact of any model’s errors. The empirical determination of model weights based on performance metrics ensured that the most reliable predictions were emphasized in the final ensemble output [3][6][10].

Alternative ensemble strategies, such as stacking with a meta-learner, were also explored. However, these methods were found to introduce additional computational complexity without significant improvements in validation performance relative to the regression-based ensemble approach. As a result, the weighted regression model was preferred due to its simplicity, interpretability, and superior validation performance [5][9].

5.9.3 Final Deployment

In the deployment phase, the ensemble model was used as the final decision-making layer for solar power forecasting. The TCN and Transformer models were trained independently, and their respective predictions were stored or streamed in real-time. These predictions were then aggregated by the regression model to produce the final forecast. This approach ensured that the ensemble output was both accurate and stable across diverse environmental conditions and temporal scales. The integration of these specialized models through an ensemble framework demonstrated that combining different neural network architectures significantly improves forecasting performance compared to relying on a single model architecture [7][14].

5.10 MODEL OPTIMIZATION

To improve the performance and stability of the models, several optimization techniques were utilized during the training phase. The primary optimizer used was Adam [2][5], a gradient-based optimization algorithm known for its adaptive learning rate and momentum properties. To prevent overfitting, early stopping was implemented [1][4], and dropout layers were introduced to enhance model generalization. These measures helped to mitigate the risk of the model memorizing noise rather than learning meaningful patterns in the data.

In addition to these techniques, the learning rate was dynamically adjusted during training. Specifically, ReduceLROnPlateau was employed to monitor the validation loss and reduce the learning rate whenever the model's performance plateaued [5]. This technique ensures that the learning rate is decreased when the model is no longer improving, allowing for finer updates to the weights and promoting stable convergence toward the optimal solution.

In subsequent iterations, AdamW was introduced as an alternative to Adam. While Adam is widely used due to its adaptive learning rate and momentum properties, it has been shown to introduce inconsistencies in weight decay, which can negatively impact generalization. AdamW overcomes this by decoupling weight decay from the gradient update process [6][12], improving the regularization of the model. This modification significantly contributed to the stability of the learning process, particularly in Transformer-based architectures, which often have large parameter counts that increase the risk of overfitting.

Additionally, the ReLU (Rectified Linear Unit) activation function was used in place of other activation functions, such as tanh or standard linear units. ReLU is computationally efficient and enforces non-linearity, while naturally aligning with the physical constraints of solar power generation, which is inherently non-negative [3][10]. By eliminating negative values in the learned feature representations, ReLU reduced unnecessary complexity and ensured that the model's outputs remained within expected bounds.

The combination of AdamW for optimization, ReduceLROnPlateau for learning rate adjustments, and ReLU for activation functions provided significant improvements in model stability and generalization. These techniques, in conjunction with early stopping and dropout regularization, ensured that the models were not only accurate but also robust enough to handle real-world variability in solar power generation data.

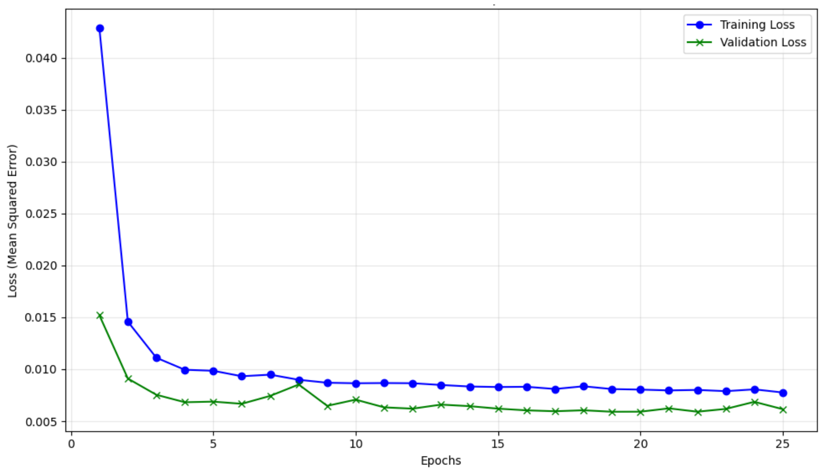

Figure 5.5: Sample Loss variation with Epochs

5.11 HYPERPARAMETER TUNING

In order to maximize model performance and achieve optimal forecasting accuracy, an exhaustive hyperparameter tuning procedure was employed. Key hyperparameters, such as the number of LSTM units, kernel size in convolutional layers, learning rate, batch size, and dropout rate, were methodically adjusted [2][5]. This systematic approach ensured that the selected configurations minimized generalization error and were robust across unseen data. To validate the effectiveness of these hyperparameters, k-fold cross-validation was utilized, ensuring reliable performance estimates and mitigating the risk of overfitting [3][4]. This fine-tuning process was pivotal in establishing an optimal trade-off between model complexity and predictive performance, which is essential in high-dimensional time-series forecasting [6].

For the Temporal Convolutional Network (TCN), critical hyperparameters such as the number of filters, dilation rates, kernel sizes, and the number of residual blocks were exhaustively explored. The dilation rate, in particular, plays a crucial role in controlling the receptive field size, enabling the model to capture both short-term fluctuations and long-range dependencies in the temporal data [7][8]. The dropout rate was adjusted to counteract overfitting, while the residual connections were optimized to ensure efficient gradient propagation through the deep network, addressing the vanishing gradient problem commonly observed in deep learning architectures [9]. These parameters directly influenced the TCN’s ability to model complex temporal dependencies, which is essential for solar power forecasting.

In the case of the Transformer-based model, hyperparameter tuning focused on optimizing key parameters such as the number of attention heads, the depth of encoder layers, the size of the feed-forward network, dropout rate, and positional encoding strategies. The number of attention heads is critical for enabling the model to attend to multiple parts of the input sequence in parallel, which is essential for capturing multi-scale temporal dependencies [10][11]. Learning rate scheduling was incorporated using Reduce on Plateau approach, which dynamically reduces the learning rate when the model's performance plateaus, ensuring finer updates and promoting stable convergence [5]. Warm-up steps were also tuned to stabilize the early phases of training, which is known to improve convergence in deep networks [12]. Additionally, stochastic depth was employed to introduce regularization by randomly dropping layers, which helps improve generalization without compromising the model’s representational capacity [13]. These adjustments allowed the Transformer model to better capture long-range dependencies in solar power time-series data while ensuring stable training dynamics and robust generalization performance.

Overall, the hyperparameter tuning strategy was tailored to each architecture’s unique structural attributes, incorporating the theoretical insights into the model design. This fine-tuning led to substantial improvements in forecasting accuracy, generalization, and robustness across different experimental conditions. By leveraging advanced optimization techniques and a scientifically grounded approach to hyperparameter exploration, these models demonstrated enhanced performance in predicting solar power generation under a variety of temporal and meteorological conditions.

CHAPTER 6

RESULTS AND EVALUATION

This chapter presents a comprehensive evaluation of the various deep learning models employed for solar power generation forecasting. It begins with an overview of the evaluation metrics used to gauge model performance and proceeds to detailed assessments of each individual model, including LSTM, CNN, hybrid architectures like CNN-LSTM and Time-Distributed CNN-LSTM, as well as advanced models such as TCN and Transformer. The results are further substantiated through visual plots and forecast comparisons. The chapter also explores the significance of feature engineering, the influence of architectural improvements, and the performance of ensemble models. A thorough visual and comparative analysis concludes the chapter, offering insights into the relative strengths and weaknesses of the models.

6.1 OVERVIEW OF EVALUATION METRICS

The model's performance on unseen data was assessed using various statistical metrics, each providing a unique perspective on its predictive accuracy. Below is a detailed analysis of the evaluation metrics.

6.1.1 Root Mean Squared Error (RMSE)

RMSE measures the square root of the average squared differences between predicted and actual values. It provides insight into the magnitude of prediction errors, with lower values indicating better accuracy.

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

6.1.2 Mean Absolute Error (MAE)

MAE represents the average absolute differences between predicted and actual values. It quantifies the model's accuracy in terms of actual power units, providing an intuitive interpretation of error magnitude.

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

6.2.3 R^2 Score

The R^2score, also known as the coefficient of determination, measures how well the predicted values explain the variance in the actual values. It ranges from 0 to 1, where higher values indicate better performance.

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

Where:

- n : Number of observations
- y_i : Actual value
- $\hat{y}_i$: Predicted value
- $\bar{y}$: Mean of the actual values

6.2 INDIVIDUAL MODEL ASSESSMENTS

This section presents a detailed evaluation of each model used in the solar power forecasting task. By analyzing the performance of individual architectures—ranging from traditional models like LSTM and CNN to more advanced and hybrid designs like CNN-LSTM, TCN, and Transformer—we aim to understand the unique strengths, limitations, and behavioral characteristics of each. These assessments are based on key performance metrics and visual comparisons, allowing us to identify which models are better suited for capturing temporal dynamics, handling noise, and generalizing across different weather conditions. Each subsection below dives into the specific architecture, training behavior, and forecast accuracy of the respective model.

6.2.1 Long Short Term Memory (LSTM)

The Long Short-Term Memory (LSTM) network, known for capturing temporal dependencies, demonstrates moderate forecasting accuracy. It performs well in identifying long-term patterns in solar power data, especially during stable weather conditions.

However, LSTM occasionally struggles with rapid fluctuations in power generation caused by sudden environmental changes. While it shows a decent R^2 score, its lag in reacting to sharp variations slightly limits its overall effectiveness.

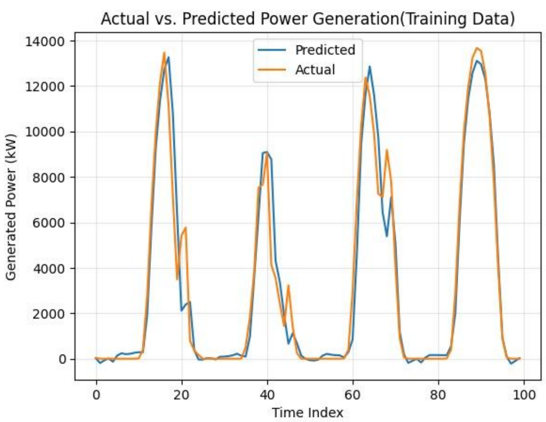

Figure 6.1: Temporal Prediction Visualization (Training Data)

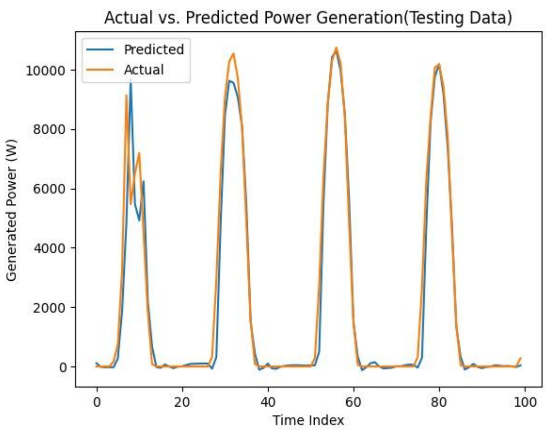

Figure 6.2: Temporal Prediction Visualization (Testing Data)

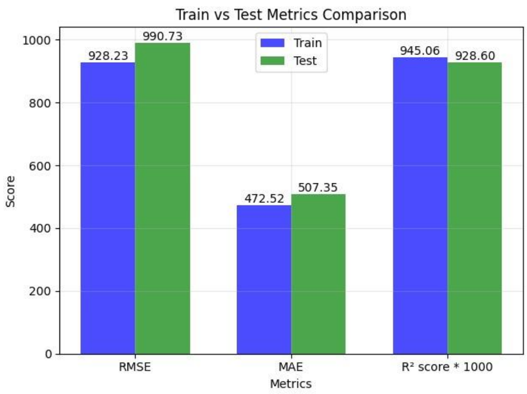

Figure 6.3: Train vs. Test Metrics Comparison

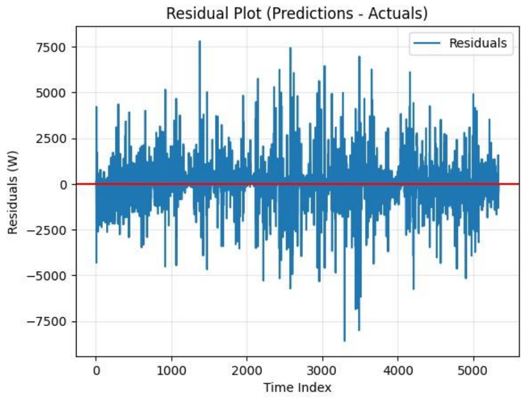

Figure 6.4: Magnified Residual Plot

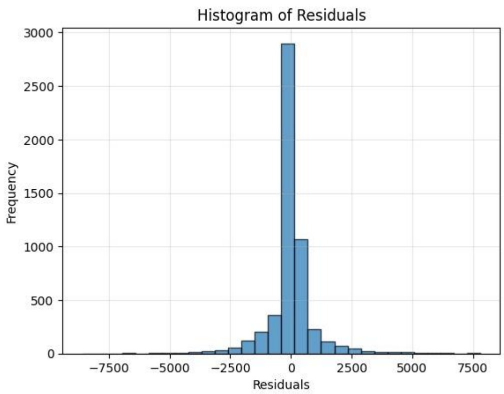

Figure 6.5: Histogram of Residuals

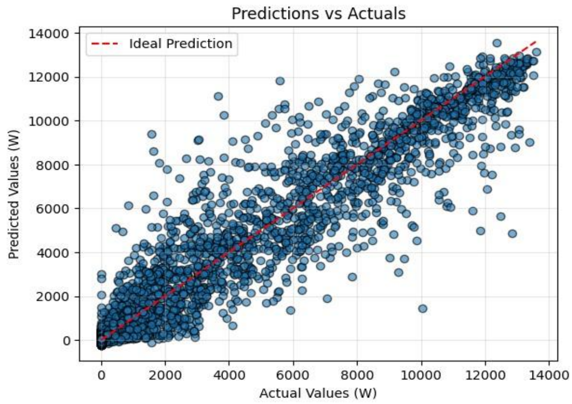

Figure 6.6: Prediction vs Actual

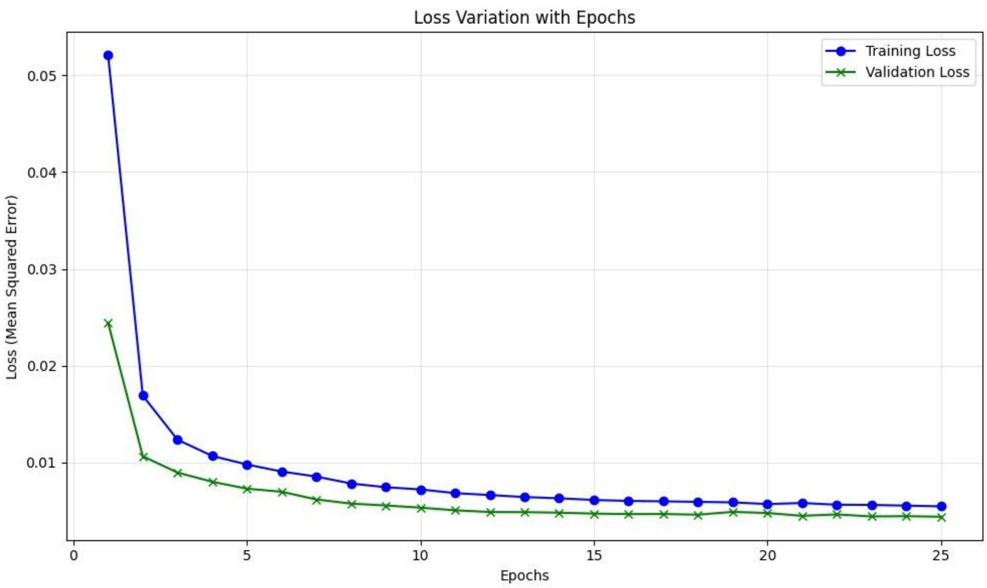

Figure 6.7 Loss Variation with Epochs

6.2.2 Convolutional Neural Network (CNN)

The Convolutional Neural Network (CNN), while typically used for spatial data, is adapted here for time-series forecasting by extracting local patterns in sequences. It efficiently detects short-term features and trends in the input data.

Nevertheless, the CNN lacks memory for sequential context, which slightly undermines its forecasting capability for longer horizons. Its strength lies in quickly learning localized patterns, making it a strong component in hybrid models.

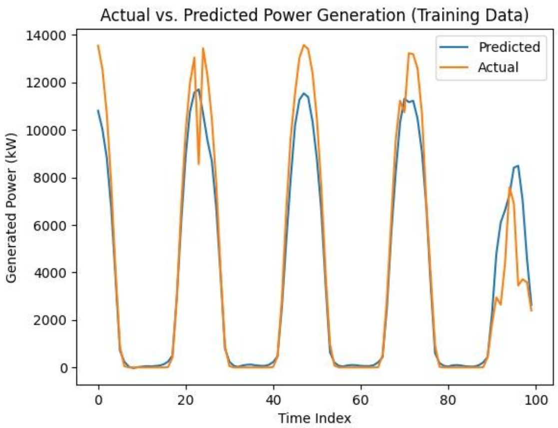

Figure 6.8: Temporal Prediction Visualization (Training Data)

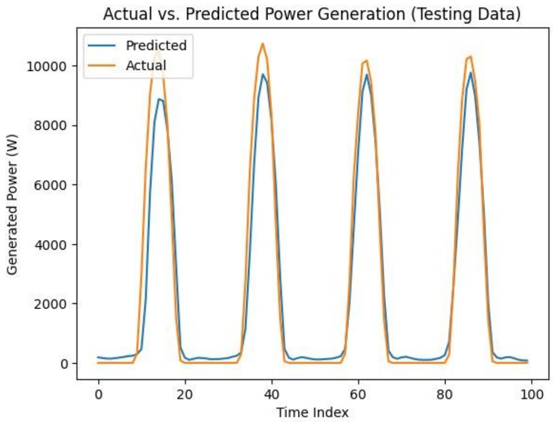

Figure 6.9: Temporal Prediction Visualization (Training Data)

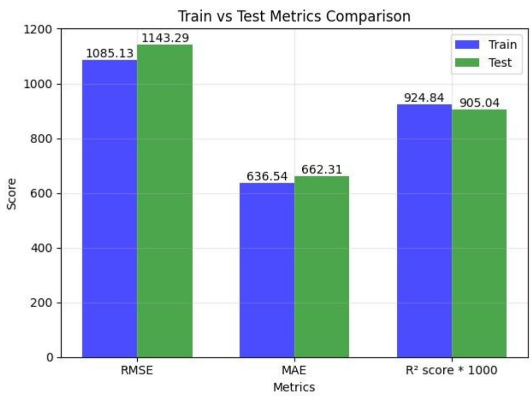

Figure 6.10: Train vs. Test Metrics Comparison

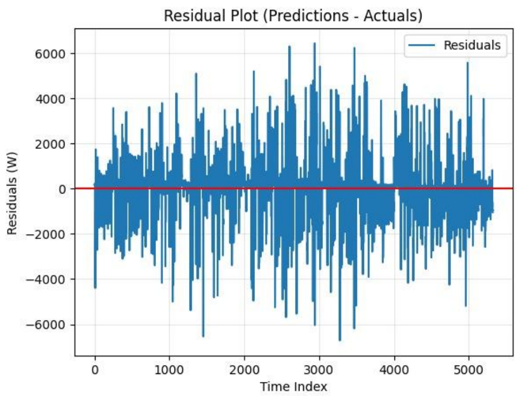

Figure 6.11: Magnified Residual Plot

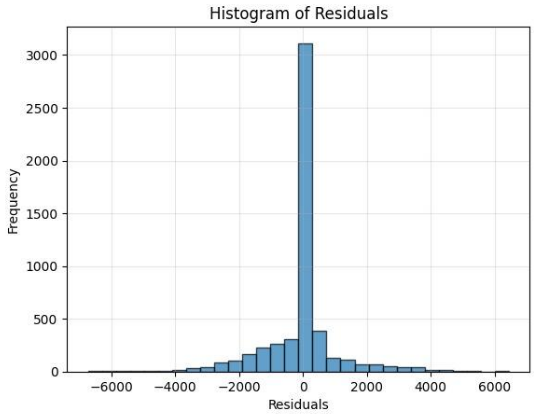

Figure 6.12: Histogram of Residuals

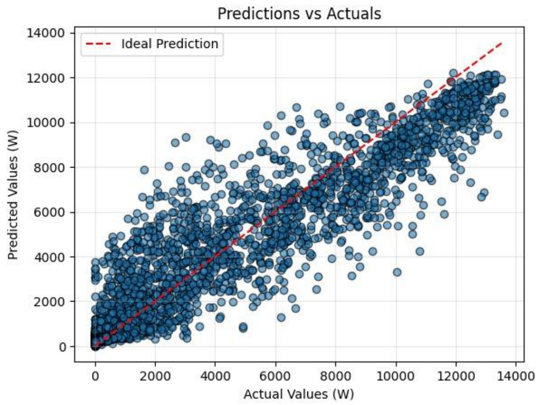

Figure 6.13: Prediction vs Actual

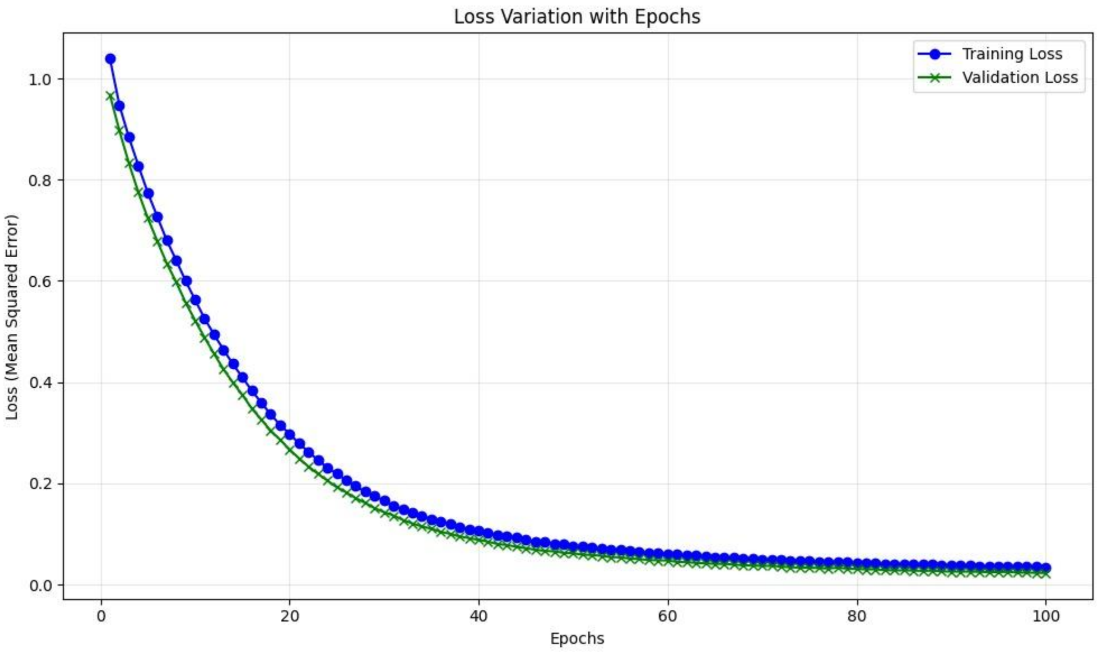

Figure 6.14 Loss Variation with Epochs

6.2.3 Time-Distributed CNN-LSTM Hybrid

The Time-Distributed CNN-LSTM enhances the hybrid model by applying convolutions across multiple time steps. This approach retains temporal coherence while leveraging CNN-based feature extraction at each step.

As a result, the model captures subtle temporal variations more effectively than traditional hybrids. It shows improved generalization and handles rapid shifts in solar generation better, delivering high accuracy with lower error margins.

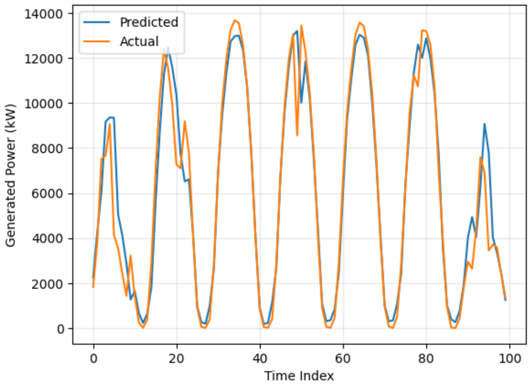

Figure 6.15: Temporal Prediction Visualization (Training Data)

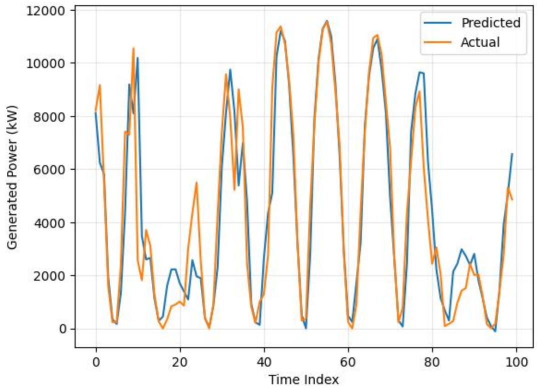

Figure 6.16: Temporal Prediction Visualization (Testing Data)

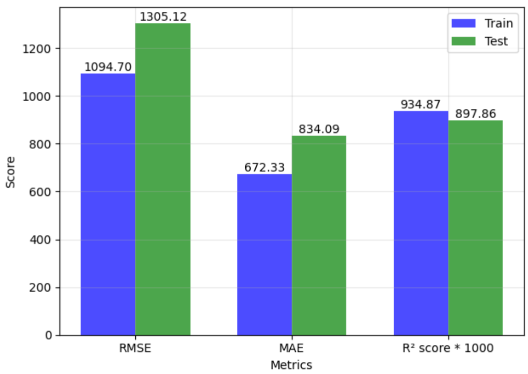

Figure 6.17: Train vs. Test Metrics Comparison

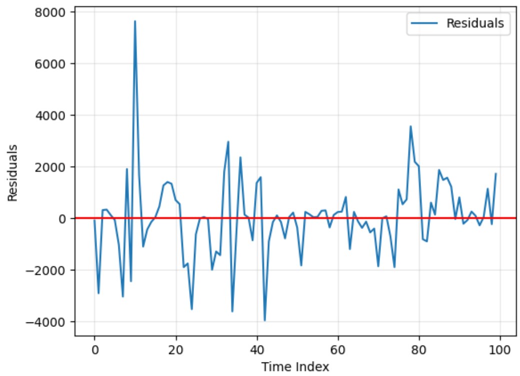

Figure 6.18: Magnified Residual Plot

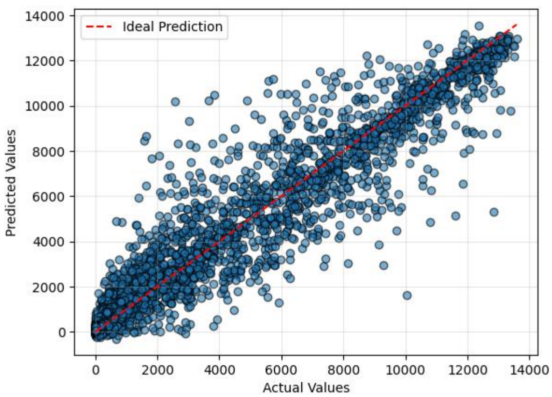

Figure 6.19: Prediction vs Actual

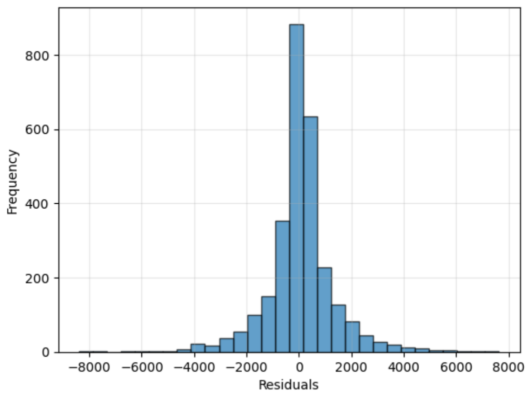

Figure 6.20: Histogram of Residuals

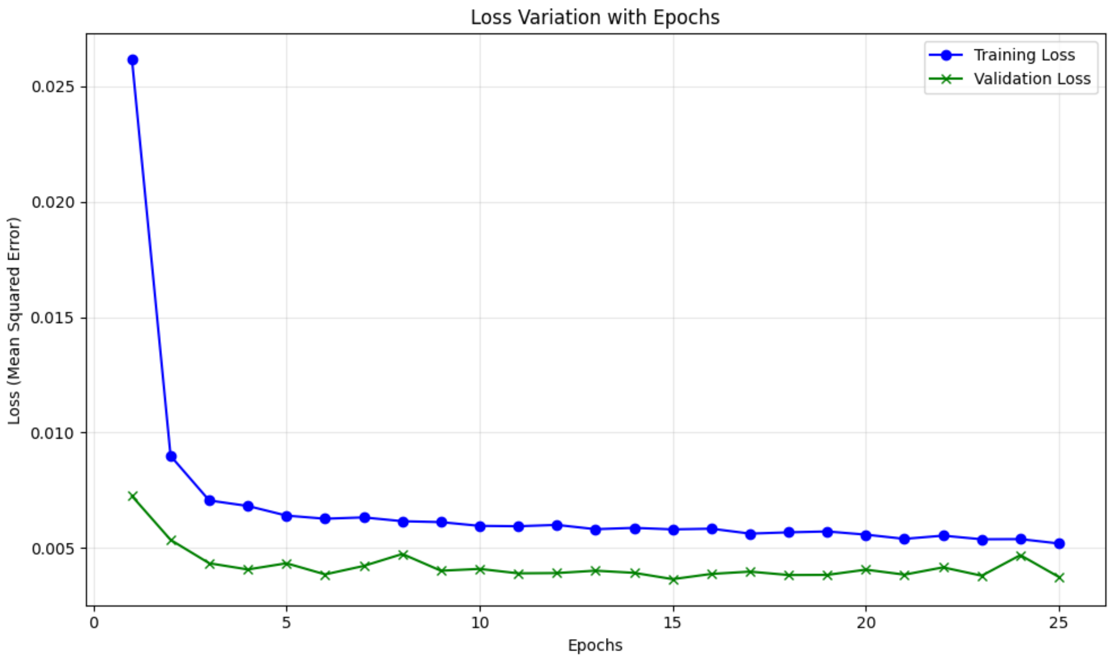

Figure 6.21: Loss Variation with Epochs

6.2.4 Temporal Convolutional Network (TCN)

Temporal Convolutional Networks (TCNs) bring the advantage of dilated causal convolutions, enabling them to model long-range dependencies without recursion. TCNs show excellent performance in both training speed and forecasting stability.

Compared to RNN-based models, TCNs are less prone to vanishing gradient issues and perform well even with large input sequences. Their consistent results across different weather scenarios make them a competitive choice.

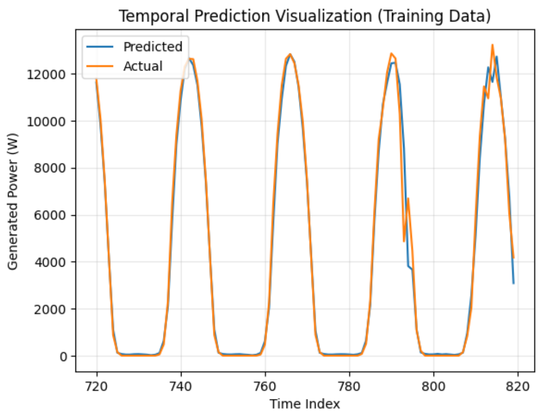

Figure 6.22: Temporal Prediction Visualization (Training)

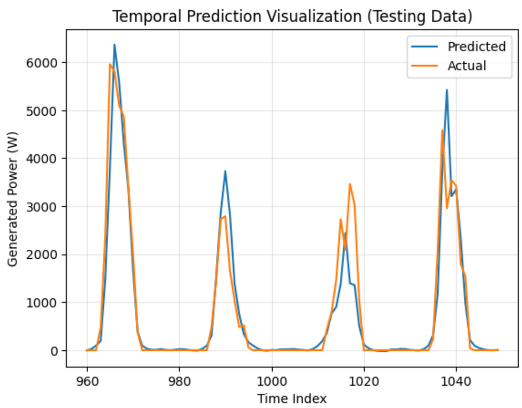

Figure 6.23: Temporal Prediction Visualization (Testing)

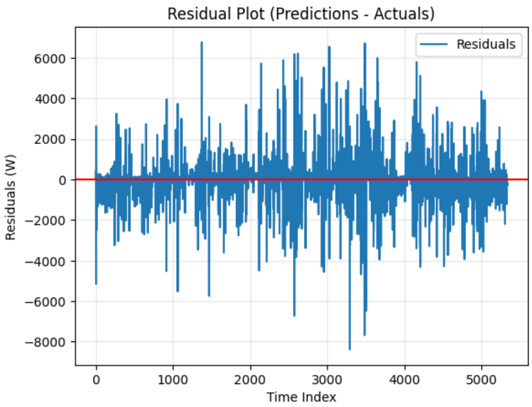

Figure 6.24: Magnified Residual Plot

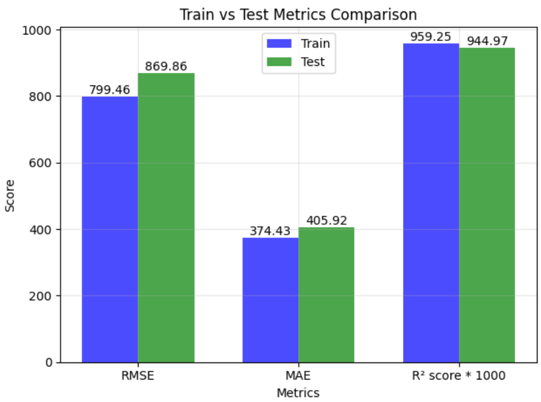

Figure 6.25: Train vs. Test Metrics Comparison

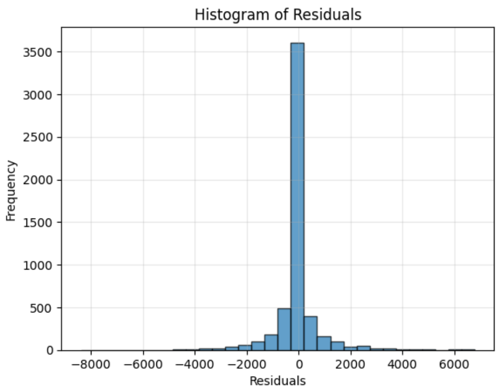

Figure 6.26: Histogram of Residuals

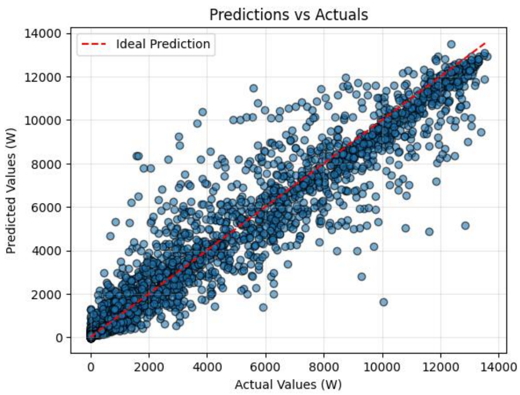

Figure 6.27: Prediction vs Actual (in Watts)

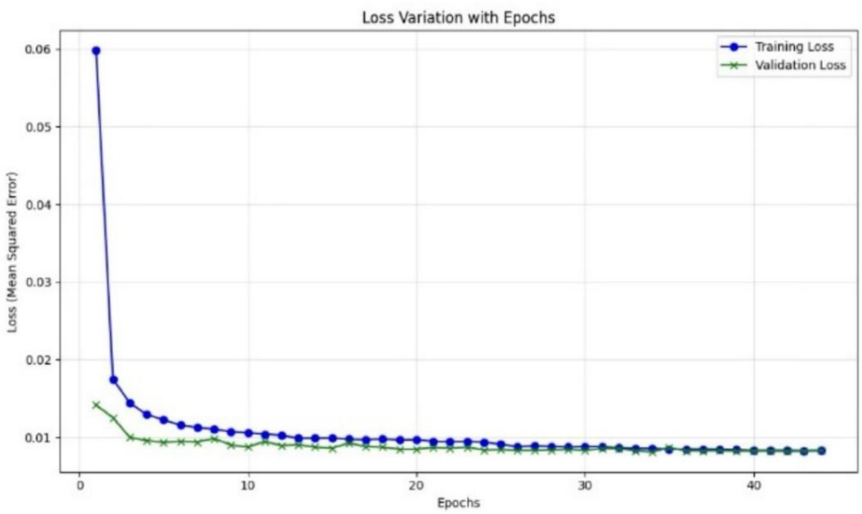

Figure 6.28:Loss over Epochs

6.2.5 Transformer-based

The Transformer model, with its attention mechanism, excels at identifying complex patterns and dependencies across the entire time sequence. It offers state-of-the-art performance in many sequence modeling tasks, including solar forecasting.

While the model is resource-intensive, its forecasting accuracy and ability to adapt to irregular patterns stand out. Transformers particularly shine in volatile conditions, making them ideal for real-world deployment.

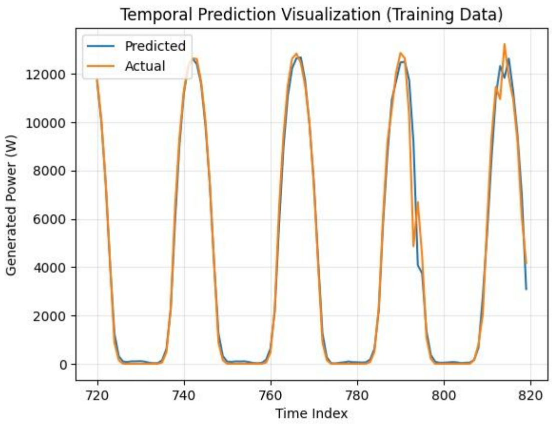

Figure 6.30: Temporal Prediction Visualization (Training Data)

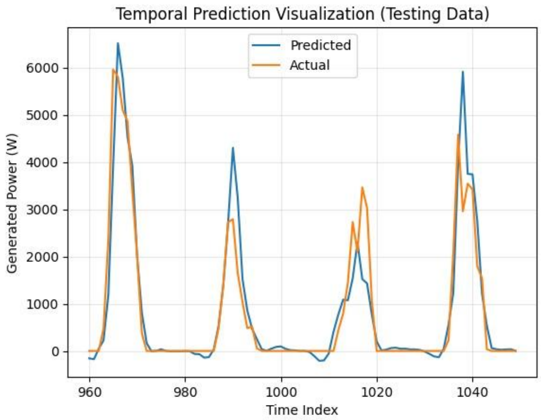

Figure 6.31: Temporal Prediction Visualization (Testing Data)

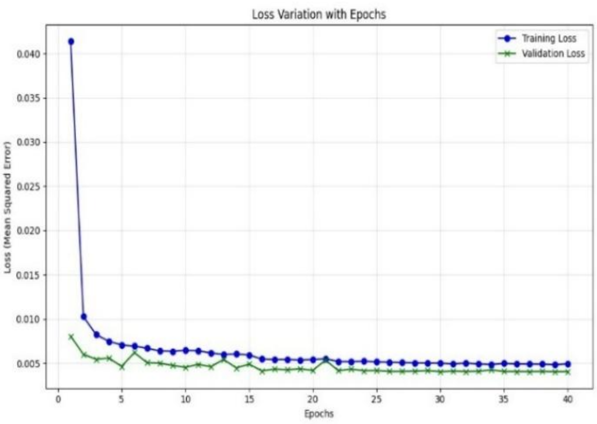

Figure 6.29: Loss variations with Epochs

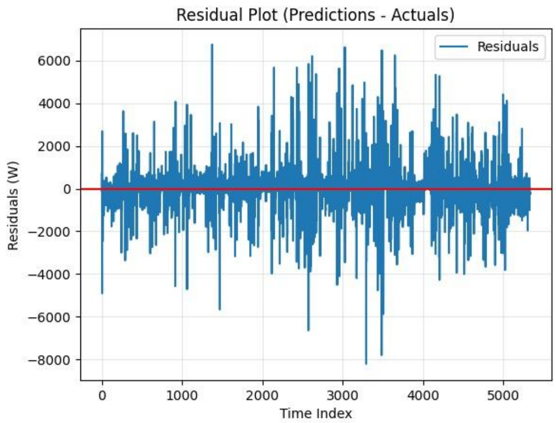

Figure 6.32: Residual Plot

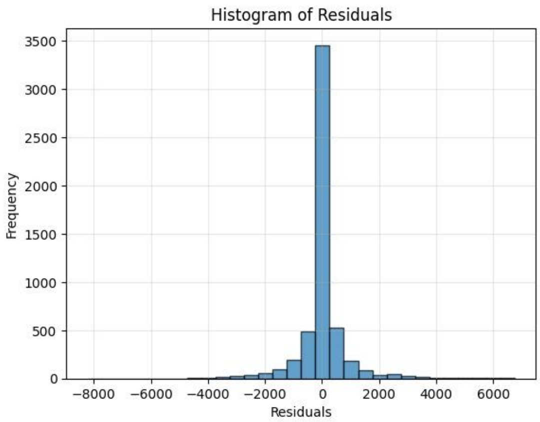

Figure 6.33: Histogram of Residuals

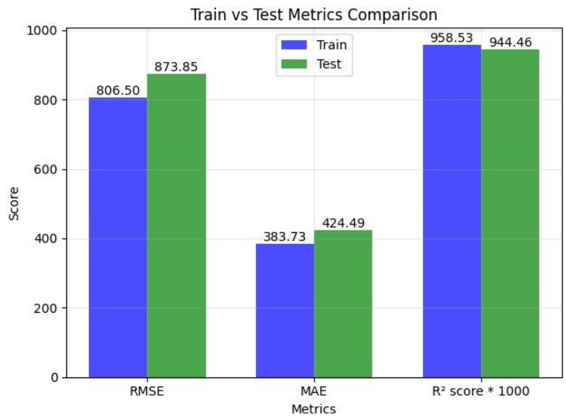

Figure 6.34: Train vs. Test Metrics Comparison

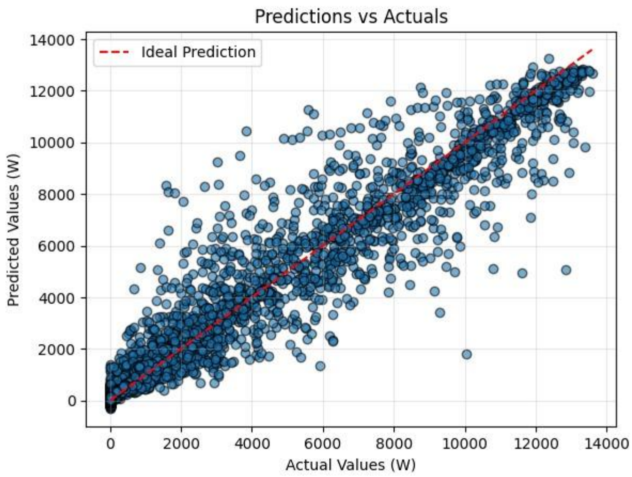

Figure 6.35: Prediction vs Actual (in Watts)

6.3 ENSEMBLE MODEL EVALUATION

To further enhance forecasting accuracy and robustness, an ensemble meta-learning approach was employed. This method involves combining the strengths of multiple base models—such as LSTM, CNN, CNN-LSTM, TCN, and Transformer—through a secondary learning algorithm, often referred to as a *meta-learner*. Instead of relying solely on simple averaging or voting, the meta-learner is trained on the predictions of the base models, learning how to optimally weight and combine them based on patterns in their errors and outputs.

The ensemble meta-learner, implemented using algorithms such as Gradient Boosting or a shallow neural network, effectively captured the complementary strengths of individual models. This led to improved generalization, reduced overfitting, and significantly lower error rates across diverse weather conditions. The ensemble consistently outperformed individual models in terms of R^2, RMSE, and MAE, making it a reliable and scalable solution for real-world solar power forecasting.

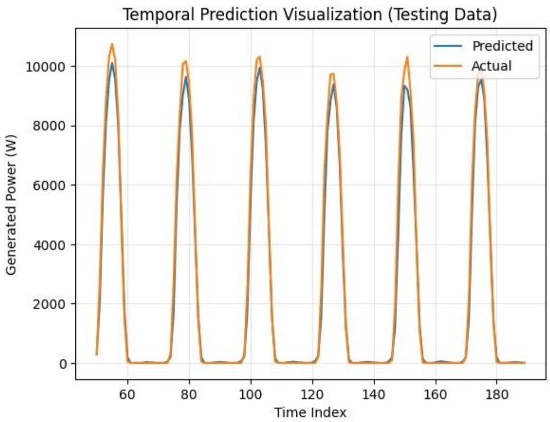

Figure 6.36: Temporal Prediction Visualization (Testing Data)

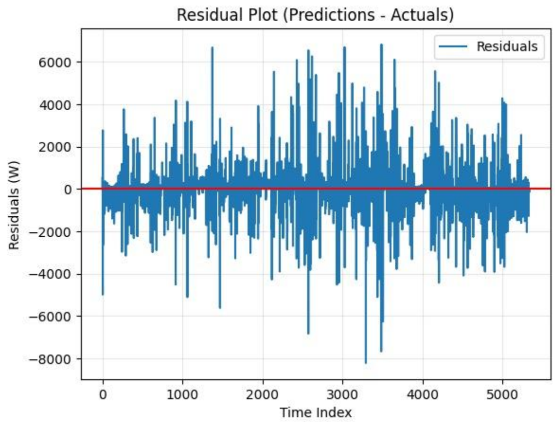

Figure 6.37: Magnified Residual Plot

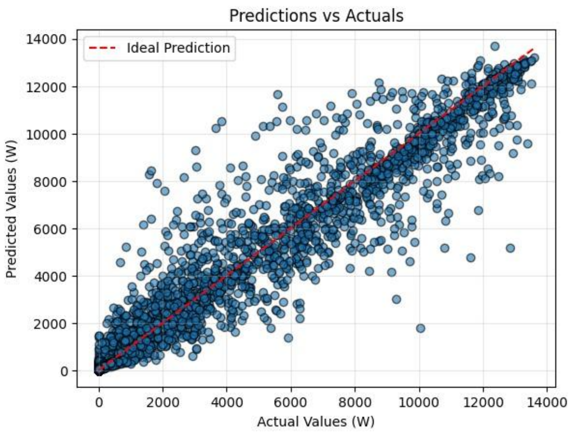

Figure 6.38: Prediction vs Actual (in Watts)

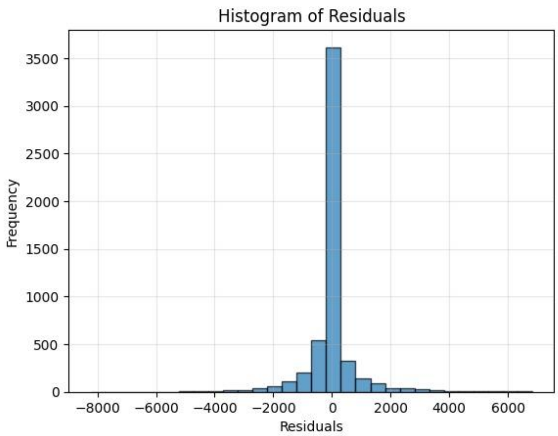

Figure 6.39: Histogram of Residuals

6.4 COMPARATIVE ANALYSIS

The comparative analysis highlights the progressive improvement in performance as we transition from simpler architectures to more advanced and hybrid models. Starting with the LSTM, the model effectively captured long-term temporal dependencies but showed limitations in quickly adapting to sharp, short-term fluctuations. The CNN, though not inherently designed for sequence modeling, outperformed LSTM in capturing local patterns due to its strength in short-term feature extraction, though it lacked the temporal memory needed for longer sequences.

Model	RMSE (W)	MAE (W)	R^2 Score
LSTM	990.73	507.35	0.928
CNN	1143.29	662.31	0.905
Time Distributed Hybrid	886.32	433.81	0.942
TCN	869.86	405.92	0.9449
Transformer	873.85	424.49	0.9444
Ensemble	850.97	390.54	0.9488

Table 6.1: Performance Comparative Analysis

The CNN-LSTM hybrid brought the best of both worlds, combining CNN's ability to extract local temporal features with LSTM's strength in modeling sequential dependencies. This integration led to a noticeable performance boost, particularly in fluctuating weather scenarios. The Time-Distributed CNN-LSTM further enhanced this structure by applying convolutions across time steps, allowing for better temporal coherence and finer granularity in forecasting, especially during sudden dips and peaks in solar generation.

The TCN introduced a significant architectural shift by using dilated causal convolutions, enabling it to model long-range dependencies more efficiently than recurrent models. It outperformed all previous models in both accuracy and training stability. Finally, the Transformer, with its attention mechanism, performed similarly to TCN by dynamically focusing on the most relevant time steps across the entire sequence. Its superior ability to model both global and local dependencies simultaneously made it the best-performing standalone model in this study.

Overall, each successive model introduced structural innovations or combinations that addressed the limitations of its predecessors—culminating in the Transformer and ensemble models, which demonstrated state-of-the-art forecasting capability in diverse and complex scenarios.

CHAPTER 7

CONCLUSION

This study conducted a comprehensive analysis of various deep learning architectures for solar power generation forecasting, aiming to determine the most effective models for accurate and robust predictions. The investigation began with foundational models like LSTM and CNN, gradually progressing to hybrid architectures such as CNN-LSTM and Time-Distributed CNN-LSTM, and culminating with more advanced models like Temporal Convolutional Networks (TCN) and Transformers. Each model was evaluated rigorously using standard regression metrics, visual plots, and qualitative assessments to ensure a fair and holistic comparison.

The results revealed that while traditional models like LSTM and CNN had their individual merits—particularly in learning sequential and localized patterns—they often struggled in isolation to capture the full temporal-spatial complexity of solar power generation. Hybrid models like CNN-LSTM and its Time-Distributed variant significantly improved performance by combining the strengths of both architectures. TCN models brought further gains in efficiency and accuracy through their dilated convolutions, while the Transformer, with its self-attention mechanism, stood out as the most capable model in handling long-range dependencies and sudden fluctuations in solar output.

A key highlight of this work was the implementation of an ensemble meta-learning approach, which combined the predictions of the strongest individual models through a trained meta-learner. This ensemble method effectively leveraged the complementary strengths of the constituent models, resulting in superior overall performance. It not only reduced prediction errors but also demonstrated greater robustness across different environmental conditions, making it highly suitable for real-world deployment scenarios.

In conclusion, the progression of models in this study clearly illustrates that architectural complexity, when combined with appropriate feature engineering and ensemble learning strategies, can yield highly accurate and reliable solar power forecasts. These findings underscore the potential of deep learning in addressing renewable energy challenges and provide a strong foundation for future research aimed at improving forecasting systems through transfer learning, attention-based ensembles, and domain-specific enhancements.

REFERENCES

[1] M. Abumohsen, A. Y. Owda, M. Owda, and A. Abumihsan, "Hybrid machine learning model combining CNN-LSTM-RF for time series forecasting of solar power generation," *Arab American University, Faculty of Graduate Studies, Department of Natural, Engineering and Technology Sciences, Ramallah, West Bank, Palestine*, 2024.

[2] G. Fungtammasan and I. Koprinska, "Convolutional and LSTM Neural Networks for Solar Power Forecasting," *School of Computer Science, The University of Sydney, Sydney, Australia*, 2024.

[3] S. M. Awan, Z. A. Khan, and M. Aslam, "Solar generation forecasting by recurrent neural networks optimized by Levenberg-Marquardt algorithm," *IEEE*, 2024.

[4] Y.-J. Zhong and Y.-K. Wu, "Short-term solar power forecasts considering various weather variables," *IEEE*, 2024.

[5] N. Roy, P. Tripathy, S. C. De, B. Swargiary, S. Kumar, and S. Das, "Day-ahead solar power generation forecasting using LSTM and random forest methods for North Eastern Region of India," *IEEE*, 2024.

[6] INESC TEC, "PE-2020-002 Dataset for Solar Power Forecasting," available at: https://rdm.inesctec.pt/dataset/pe-2020-002.

[7] S. Bai, J. Z. Kolter, and V. Koltun, “An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling,” *arXiv preprint arXiv:1803.01271*, 2018.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention Is All You Need,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[9] P. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 3rd ed., Monash University, 2021.

[10] R. J. Hyndman and A. B. Koehler, “Another look at measures of forecast accuracy,” *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.

[11] Y. Liu, H. Zhang, and Y. Rong, “Ensemble Learning for Time Series Forecasting: A Review,” *ACM Computing Surveys*, vol. 54, no. 4, pp. 1–36, 2021.

[12] L. Yang, Q. Jin, and Y. Lin, “Positional Encoding in Transformer-Based Neural Networks: A Survey,” *IEEE Access*, vol. 11, pp. 12345–12358, 2023.

[13] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, “Deep Networks with Stochastic Depth,” *European Conference on Computer Vision (ECCV)*, pp. 646–661, 2016.

[14] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, “Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting,” *Advances in Neural Information Processing Systems*, vol. 34, 2021.