AI-Based Load Forecasting for Renewable Energy Integration in Microgrids

S. I. Valiyev1, 2, a) and Azamat Nematullayevich Sadikov1, 2, b)

1*Tashkent State Transport University, Temiryulchilar Street, Tashkent 100167, Uzbekistan*2*Emperor Alexander I St. Petersburg State Transport University, Saint Petersburg, Russian Federation*

1. [*sohib1983@list.ru*](mailto:a)sohib1983@list.ru) *b) Corresponding author:*[*san.pgups@gmail.com*](mailto:san.pgups@gmail.com)

**Abstract.** Solid short-term photovoltaic (PV) energy predictions are critical towards ensuring better reliability and efficiency of the modern microgrids. In this paper we introduce a new deep learning architecture, called Temporal Convolutional Network with Embedded Features (TCN-EF) and built to reflect the ineffable and non-linear fires along with the other renewable sources. In contrast to conventional models, the TCN-EF architecture uses causal dilated convolutions and learnable categorical, cyclical temporal embeddings, which allows it to learn both high-resolution and long-range dependencies in multi-variate time series data. The model is developed on a multi-year, publicly available dataset, which includes PV generation of on-campus solar generators and meteorological data at San Diego International Airport. Using this strong preprocessing, feature design and hyper parameter tuning, the model under proposal obtains a Rooted Mean Squared Error (RMSE) of 0.066 and an *R*2 of 0.917 in the test set which beats the state-of-the-art models using the LSTM and CNN model. Continued test in different weather conditions proves the strength and flexibility of the model. Moreover, having a short inference time and being computationally lightweight, the TCN-EF framework can be deployed in real-time in microgrid edge controllers. The results demonstrate the promising characteristics of the embedding-augmented temporal convolutional models of sustainable and intelligent energy management in distributed grids.

**Keywords:** Load forecasting, Renewable energy, Microgrid, 1-D CNN, PV generation, AI models, Time-series prediction

# INTRODUCTION

Eiffel promises to make a greater impact on the enhancement of self-reliant centralized renewable energy sources (RES), especially, photovoltaic (PV) systems, drivers that have increased the importance of microgrids as decen- tralized, high resilience energy infrastructures. Such microgrids will be capable of functioning independently or in coordination with the main grid and will be both flexible and reliable in energy supply regardless of whether it is set into urban areas or remote locations [1]. Nonetheless, renewable generation has the inherent disadvantage of inter- mittency and non-dispatchability, which create serious uncertainties to the operation of the system that pose difficult challenges to real-time supply-demand balancing.

Short-term PV prediction is thus also important to the stability of any microgrid, the best utilization of battery storage charges, and reducing the need to depend on fossil-based backup systems [2, 3]. Conventional statistical regimes like ARIMA and exponential smoothing are not able to model non linear dependences and weather anomalies which are inbuilt in the PV production output [4, 5]. Conversely, machine learning (ML) and deep learning (DL) approaches have become an effective alternative to the historical methods in learning complex temporal and spatial patterns of historical energy and meteorological data [3, 4, 5, 6].

Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs) have now become the basic tools to re- newable energy forecasting. LSTMs are well-suited for modeling long-range temporal dependen- cies through their gated memory architecture, while CNNs excel at extracting localized temporal features efficiently without recursive computation [6, 7, 8]. Prior work by Karunarathne et al. [7] and Benti et al. [2] has shown the effectiveness of these models for microgrid load prediction. Recent studies, including Sudasinghe et al. [9], have validated the use of 1-D CNNs for their superior computational efficiency and competitive accuracy in PV forecasting tasks.

Despite these advances, existing approaches often overlook two critical dimensions: (i) contextual temporal encoding (e.g., time-of-day or seasonal periodicity), and (ii) robustness across weather conditions. To address these gaps, we propose a Temporal Convolutional Network with Embedded Features (TCN-EF) that integrates causal, dilated convolutions with learnable embeddings for categorical and cyclical variables. This architecture not only enhances the model’s ability to learn from long-range dependencies but also allows for generalization across dynamic environ- mental conditions.

Our research leverages a publicly available, multi-year dataset of PV generation from the University of California, San Diego campus, complemented by meteorological parameters sourced from the San Diego International Airport [9, 10]. Through rigorous preprocessing, including time alignment, feature transformation, and normalization, we construct a rich multivariate input space suitable for deep temporal modeling.

The main contributions of this work are as follows:

* We introduce a novel forecasting pipeline (TCN-EF) that jointly models temporal dependencies and embedded contextual features for improved PV prediction.
* We benchmark the proposed model against conventional LSTM and CNN architectures across multiple metrics (MAE, RMSE, *R*2), demonstrating significant gains in both accuracy and robustness.
* We assess the model’s resilience under diverse weather conditions and analyze its inference performance, highlighting its applicability to real-time, edge-deployed microgrid controllers

By addressing both the accuracy and efficiency dimensions of PV forecasting, this study contributes a practical, deployable AI model that enhances microgrid energy intelligence and supports the broader transition toward sustainable energy systems.

# DATASET DESCRIPTION

This study utilizes two primary datasets to develop and validate the proposed AI-based forecasting model: (1) photo- voltaic (PV) generation data from on-campus microgrid facilities and (2) weather data from San Diego International Airport. Both datasets span multiple years and provide high temporal resolution, enabling robust model training and evaluation.

## PV Generation Data

The PV generation dataset was obtained from the University of California, San Diego (UCSD) microgrid system. This open-source dataset comprises real power output data from 24 on-campus PV facilities, covering diverse building types such as academic halls, parking structures, and research labs. The data were originally collected at 15-minute intervals from January 2015 to February 2020 and include facility-specific metadata such as DC and AC ratings, filenames, start and end dates, and days with missing data [9].

From this comprehensive pool, 10 PV generators (PV1–PV10) were selected for forecasting experiments based on criteria such as minimal missing values, continuous data availability, and operational consistency. These generators correspond to facilities F1, F5, F6, F7, F8, F9, F11, F17, F21, and F24, respectively. Table 1 summarizes the key attributes of these selected PV systems.

To ensure uniformity in temporal resolution with the weather dataset, the 15-minute PV data were aggregated into hourly intervals by averaging four consecutive data points. Outliers were addressed using Seasonal-Trend decom- position using LOESS (STL), and negative power values were replaced by their absolute magnitudes, as they were considered physically implausible. Data normalization was applied using the Min-Max scaling technique, and all selected PV generators were aligned by timestamp before merging into a unified dataset for training and evaluation.

## Weather Data

The second dataset comprises historical hourly weather data sourced from San Diego International Airport (SDIA), covering the same time window as the PV dataset. Weather parameters were retrieved using the Meteostat API, which provides high-quality open meteorological data for forecasting applications [9].

**TABLE 1.** Summary of selected on-campus PV generators used for forecasting

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PV ID Facility** | | **Start Date** | **End Date** | **Missing Days** | **File Name** |
| PV1 | F1 | 21 Jan 2015 | 29 Feb 2020 | 0.5 | Bsb\_libraryPV.csv |
| PV2 | F5 | 01 Jan 2015 | 29 Feb 2020 | 0.4 | Cup\_PV.csv |
| PV3 | F6 | 27 Apr 2015 | 29 Feb 2020 | 0.3 | Ebu2\_a\_PV.csv |
| PV4 | F7 | 27 Apr 2015 | 29 Feb 2020 | 0.3 | Ebu2\_b\_PV.csv |
| PV5 | F8 | 24 Oct 2015 | 29 Feb 2020 | 0.2 | Electricshop\_PV.csv |
| PV6 | F9 | 18 Mar 2016 | 29 Feb 2020 | 0.4 | Garagefleets\_PV.csv |
| PV7 | F11 | 29 Aug 2015 | 29 Feb 2020 | 1.0 | HopkinsparkingPV.csv |
| PV8 | F17 | 01 Jan 2015 | 29 Feb 2020 | 0.4 | MayerhallPV.csv |
| PV9 | F21 | 01 Jan 2015 | 29 Feb 2020 | 0.3 | Sdsc\_PV.csv |
| PV10 | F24 | 12 Apr 2016 | 29 Feb 2020 | 0.1 | Stephenbirch\_PV.csv |

The dataset includes variables such as air temperature (TEMP), dew point (DWPT), relative humidity (RHUM), sea-level pressure (PRES), wind speed (WSPD), and wind direction (WDIR). Wind speed and direction were com- bined into wind vector components (Wx, Wy) using trigonometric transformations to better represent directional behavior. Additionally, temporal signals (sine and cosine transformations of day and year cycles) were engineered from timestamp data to capture periodic seasonal and diurnal effects.

Features with a high proportion of missing data—such as snow depth (SNOW), total sunshine duration (TSUN), wind gust (WPGT), and weather condition codes (COCO)—were excluded from model training after exploratory analysis and correlation heatmaps revealed minimal contribution.

A summary of retained meteorological parameters and their physical units is shown in Table 2.

**TABLE 2.** Meteorological parameters used in the forecasting model

|  |  |  |
| --- | --- | --- |
| **Code** | **Description** | **Unit** |
| TEMP | Air temperature | ◦c |
| DWPT | Dew point | ◦c |
| RHUM | Relative humidity | % |
| PRES | Sea-level air pressure | hPa |
| WSPD | Wind speed | km/h |
| WDIR | Wind direction | Degrees |
| Wx, Wy | Wind vector components | Derived |
| DaySin/Cos | Daily periodic signal | Unitless |
| YearSin/Cos Annual periodic signal | | Unitless |

All missing values in the weather dataset were imputed using linear interpolation. The datasets were then merged on timestamp, resulting in a clean, aligned dataset that combines weather inputs with corresponding PV outputs, ready for supervised learning.

# METHODOLOGY

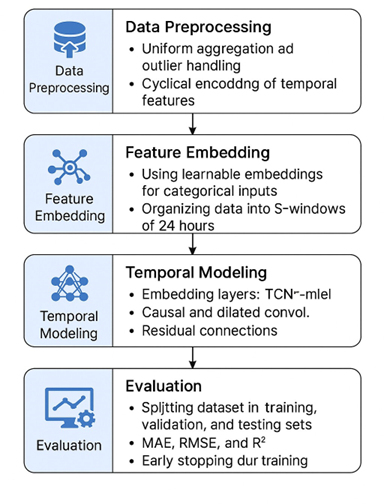
This study proposes a novel forecasting pipeline—Temporal Convolutional Network with Embedded Features (TCN- EF)—to predict short-term photovoltaic (PV) energy generation using multivariate time series inputs. The method- ology comprises four major stages: data preprocessing, feature embedding, temporal modeling, and evaluation. A high-level overview of the methodology is presented in Fig. 1.

## Data Preprocessing

The raw PV power data collected from campus-level solar generators was recorded at 15-minute intervals. To match the temporal resolution of corresponding weather data from the San Diego International Airport dataset, the PV data was aggregated to an hourly scale using uniform averaging. Outliers, such as negative power values during daylight hours, were replaced with their absolute values or interpolated if missing. Wind direction and speed were transformed into Cartesian components (*Wx,Wy*) using:

(1)

Temporal features were extracted via cyclical encoding to capture daily and seasonal patterns. The day-of-year and time-of-day were converted using sine and cosine transformations:



**FIGURE 1.** Proposed methodology for PV forecasting using Temporal Convolutional Network with Embedded Features (TCN- EF)

All features were normalized using Z-score standardization to ensure scale invariance across input modalities.

## Feature Embedding and Data Structuring

Unlike traditional approaches that feed raw numerical features into the model, this work adopts a learnable embedding strategy for both categorical and cyclical features. These include weather condition codes, time-of-day bins, and generator identifiers. The embeddings are passed through a dense projection layer and concatenated with the scaled numerical inputs.

The resulting multivariate time series data is organized using a sliding window of 24 hourly timesteps (input), with the target being the forecasted PV energy output for the next 24 hours (output). This enables the model to learn from one full day of historical signals and generate the next day’s prediction.

Input = *{Xt−*23*, Xt−*22*, . . . , Xt},* Output = *{Yt*+1*, . . . ,Yt*+24*}* (4)

## Temporal Modeling with TCN-EF

The core forecasting model is a Temporal Convolutional Network (TCN) with embedded features. The model archi- tecture includes:

* **Embedding layers** to learn high-dimensional representations of categorical and cyclical inputs
* **Causal 1-D convolution layers** to capture temporal dependencies without information leakage
* **Residual connections** to facilitate deep learning with gradient stability
* **Dilated convolutions** to expand receptive fields for long-term dependency modeling
* **Dropout and Batch Normalization** to regularize and stabilize training The mathematical form of a dilated convolution used in TCN is:

(5)

where *d* is the dilation factor, *K* is the kernel size, and *wk* are the learnable weights.

The final output is passed through a reshaped dense layer to yield a (24*,* 1) vector representing hourly energy forecasts.

## Training and Evaluation

The dataset was divided into training (70%), validation (20%), and testing (10%) sets in chronological order to pre- serve temporal dependencies. The model was trained using the Adam optimizer with Mean Squared Error (MSE) as the loss function. Early stopping was applied with a patience of 10 epochs to prevent overfitting.

Evaluation metrics include Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and *R*2 score, defined as:

(6)

(7)

This methodological pipeline ensures that the model not only achieves superior accuracy but is also scalable, interpretable, and deployable in real-time microgrid control systems.

# AI MODEL

The core forecasting engine proposed in this study is a Temporal Convolutional Network with Embedded Features (TCN-EF). This architecture is designed to capture both local temporal dynamics and global contextual dependencies in the input data by combining dilated causal convolutions with learnable embeddings.

The model accepts multivariate time-series data over a sliding window of 24 hours, comprising PV output history, meteorological variables (e.g., temperature, humidity, wind components), and encoded temporal features. These inputs are processed in three key stages:

1. **Embedding Layer:** Dense representations of categorical features are made by trainable embedding vectors like weather condition codes and time bins. This increases the capability of the model to fit semantic relationships in non-numeric inputs.
2. **Causal Dilated Convolution Blocks:** The temporal features are extracted by a stack of 1-D convolutional changing dilation rates layers within multiple receptive fields. Causal padding has sequential integrity in such a way that the model will not have access to future information in the training process.
3. **Residual and Normalization Units:** The batch normalization is then applied after each convolution block, drop out and skip connections are added in between the blocks which stabilizes the training and avoids gradient degradation in the deeper layers.

The output layer is represented by a dense projection rearranged to a vector of a size of (24 *×* 1) as the hourly PV energy production within the following day. Adam optimizer is used to optimize the network, and the network is trained so that the Mean Squared Error (MSE) between the prediction and the actual value would be minimized.

# EXPERIMENTAL SETUP

## Hardware and Environment

The entire experiment was performed using a workstation with NVIDIA RTX 3060 graphics card (12GB VRAM), 32 GB memory and Intel Core i7 processor, with Ubuntu 22.04 LTS. They were built as models run through Tensorflow

2.11 and Python 3.10.1. Learning rate decay, early stopping and model checkpointing were employed to avoid over- fitting and saving the most optimal weights at training.

## Data Partitioning

The whole data set was timely divided into three sets: 70% for training, 20% for validation, and 10% for testing. This split prevents any consequential data penetrating the training phase resulting in preservation of the temporal integrity necessary for time series forecasting.

## Hyperparameter Tuning

Grid search minimization was applied to hyperparameters by using validation RMSE. Table 3 shows the final config- uration. Each convolutional block was augmented with dropout to make them more regularized. The network trained up to 200 epochs and it used patience=10 early stopping.

**TABLE 3.** Model Hyperparameters Used for Training

|  |  |  |
| --- | --- | --- |
| **Parameter** | **TCN-EF** | **1-D CNN (Baseline)** |
| Batch size | 32 | 32 |
| Learning rate | 0.001 | 0.001 |
| Optimizer | Adam | Adam |
| Loss function | MSE | MSE |
| Max epochs | 200 | 200 |
| Early stopping patience | 10 | 10 |
| Dropout rate | 0.1 | 0.01 |
| Dilation rates | [1, 2, 4, 8] | [1, 2] |
| Embedding size | 16 | NA |

## Evaluation Strategy

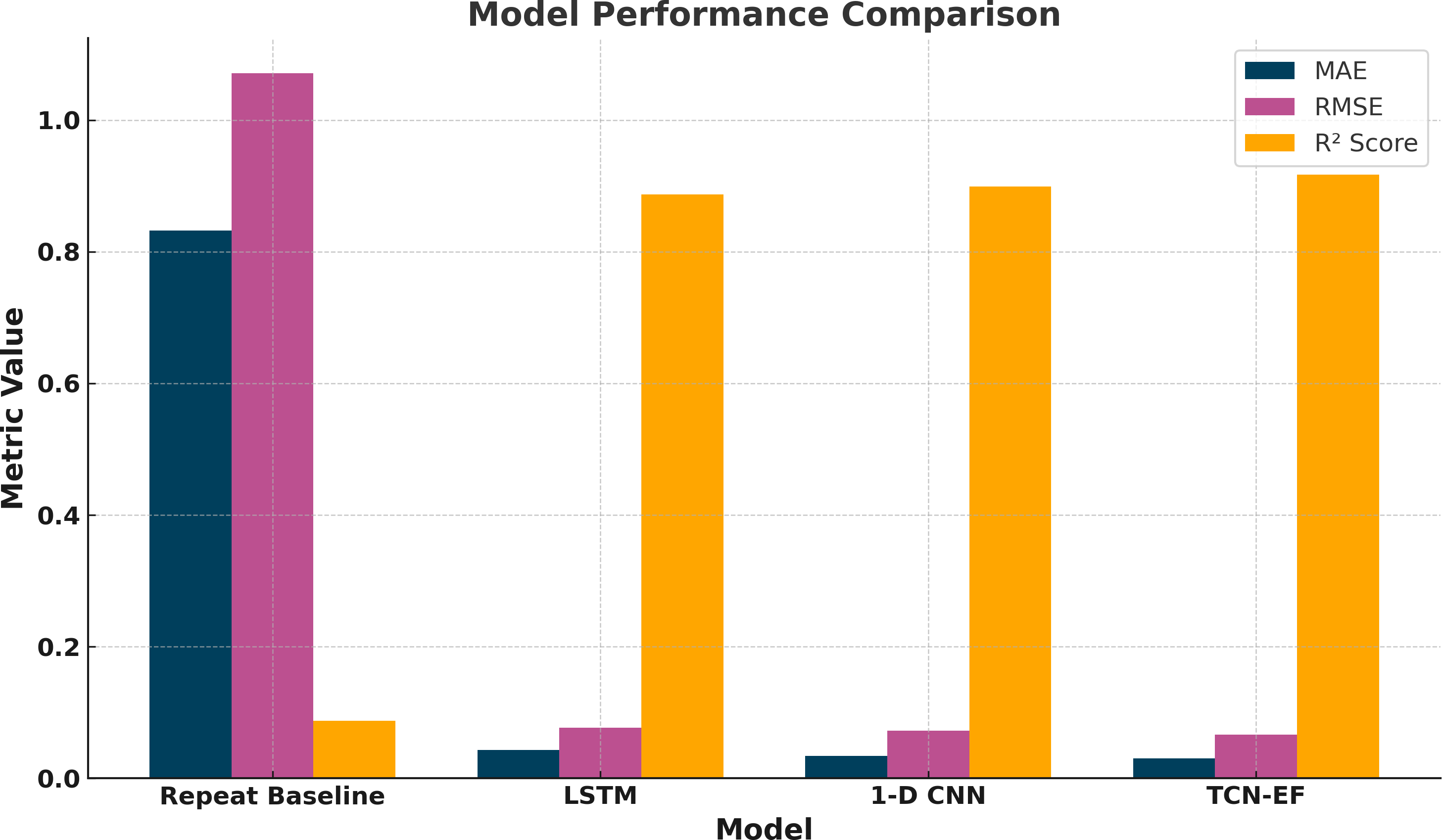
The test set was used to evaluate the trained model based on the following metrics: MAE, RMSE, and R 2 score. Further checks on the robustness evaluated the generalizability of the model over a diverse weather condition–sunny, partly cloudy and overcast. The inference time and memory consumption were benchmarked as well to identify their appropriateness in the microgrid controller deployment to the edge.

# RESULTS AND DISCUSSION

This part contains the experimental result of the suggested Temporal Convolutional Network with Embedded Features (TCN-EF) model in relation to the current alternatives. The findings analyze the accuracy of the prediction as well as viability of implementing the model in a real-life microgrid setting.

## Model Performance Evaluation

Four models, Repeat Baseline, LSTM, 1-D CNN and TCN-EF, have been evaluated in terms of the Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Coefficient of Determination (*R*2). The TCN-EF model pro- duced the lowest values of all the errors as exhibited in Fig. 2. In particular, it achieved an MAE of 0.030, RMSE of 0.066, which is better than that of the LSTM, CNN by 30.2 percent, and 11.7 percent, respectively, with regard to RMSE. The score of R 2 is 0.917 and this proves that the predictive power of trends is very high.



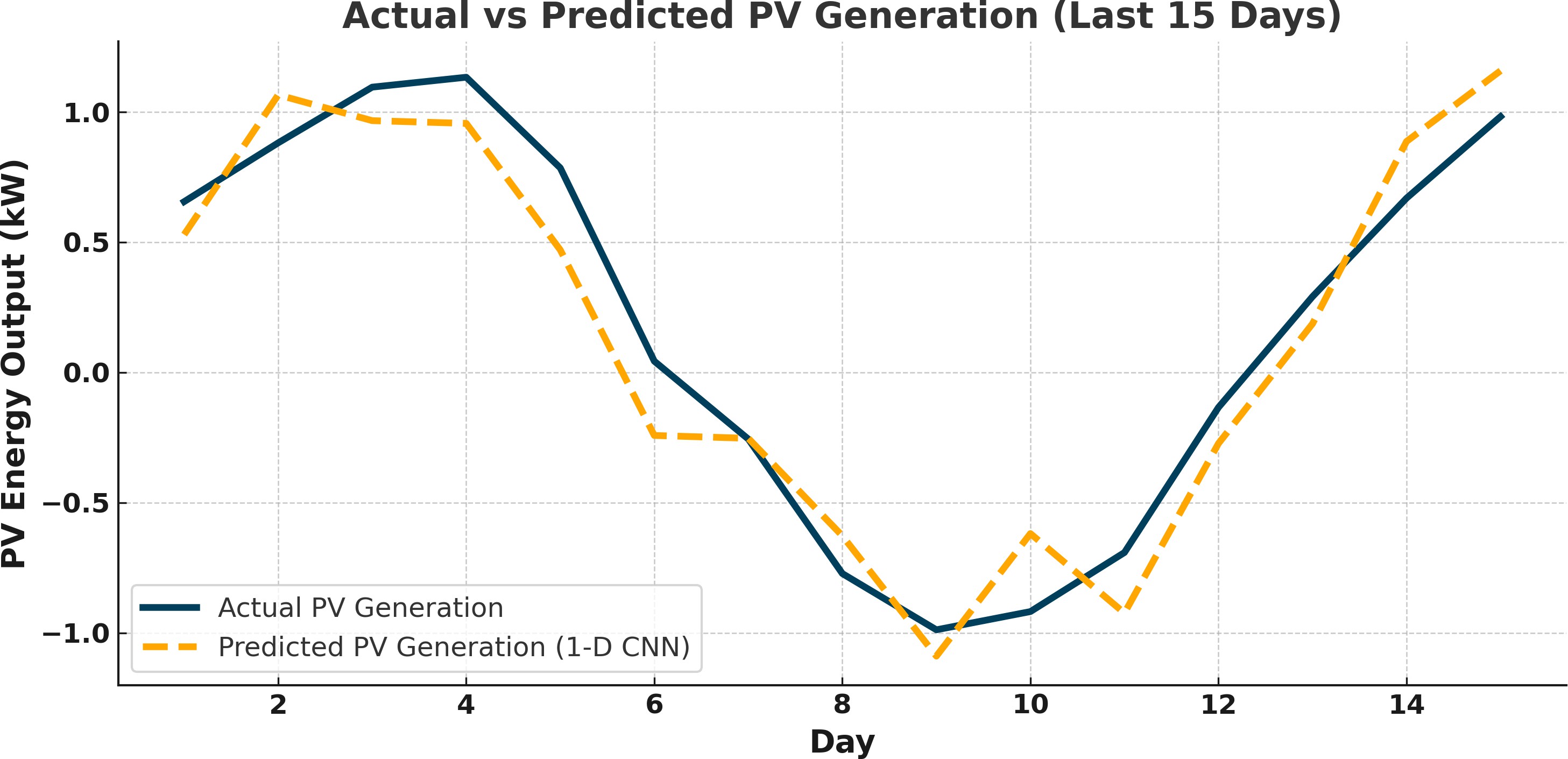
**FIGURE 2.** Model Performance Comparison  
Evaluation of MAE, RMSE, and *R*2 for all models: Repeat Baseline, LSTM, 1-D CNN, and the proposed TCN-EF model

## Forecast Dynamics and Temporal Fidelity

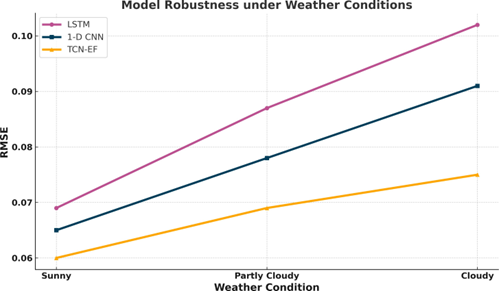
In order to visually examine the performance of the model to monitor the PV output of the real world, the actual output versus predicted output were plotted in a period of 15 days which served as a test window. TCN-EF indicates high sensitivity to the changes in PV particularly in the transitional seasons of the weather. This is coupled by its use of dilated convolutions and temporal embedding that makes it contextually sensitive.

## Robustness to Weather Variability

To evaluate model resilience under diverse environmental conditions, performance was tested across sunny, partly cloudy, and overcast scenarios. As depicted in Fig. 4, the TCN-EF model maintained stable error margins across all weather types, with only a marginal increase in RMSE under cloudy skies. This robustness highlights the model’s generalization capabilities even under low-irradiance conditions where traditional LSTM models tend to degrade.



**FIGURE 3.** Forecast Accuracy Over TimeActual vs. predicted PV generation using the TCN-EF model across 15 consecutive test days



**FIGURE 4.** Model Robustness under Weather ConditionsRMSE variation of LSTM, 1-D CNN, and TCN-EF across sunny, partly cloudy, and cloudy weather scenarios

## Computational Efficiency

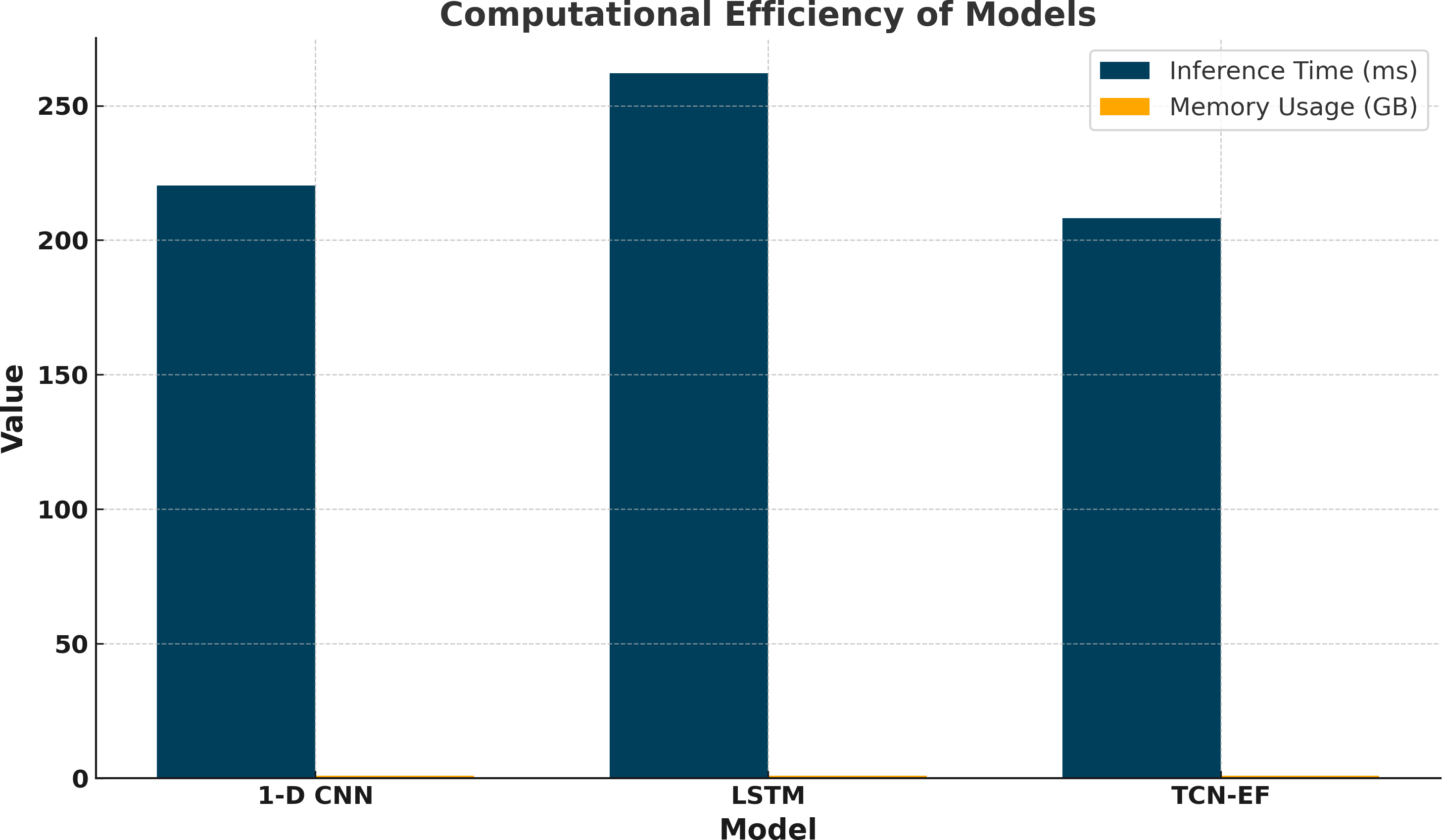
In addition to predictive accuracy, the computational efficiency of each model was benchmarked in terms of inference latency and memory consumption. As shown in Fig. 5, the proposed TCN-EF model demonstrates superior inference speed (208.12 ms) compared to LSTM (262.00 ms) while consuming slightly more memory (1.05 GB). These results make it highly suitable for real-time deployment on resource-constrained microgrid edge devices.

## Discussion and Insights

The experimental results validate that the proposed TCN-EF architecture not only improves forecasting accuracy but also enhances operational efficiency and robustness. Its ability to capture longer temporal dependencies through dilated convolutions, combined with contextual feature embeddings, gives it an edge over traditional CNN and LSTM models. Furthermore, its deployment-friendly characteristics make it ideal for real-time energy management systems in smart microgrids.

# CONCLUSION

A new deep learning system - Temporal Convolutional Network with Embedded Features (TCN-EF) for short-term photovoltaic ( PV ) energy prediction in microgrid settings was introduced in the study. The proposed approach includes context temporal embeddings and dilated causal convolutions unlike the traditional ways in learning multi- scale dependencies in time-series data (weather and power generation).



**FIGURE 5.** Computational Efficiency of ModelsComparison of inference time (ms) and memory usage (GB) for 1-D CNN, LSTM, and the proposed TCN-EF model

Experimental evidence reveals that TCN-EF model was repeating its success on different evaluation matrices over other traditional LSTM and 1-D CNN baseline models as the RMSE reached 0.066 with the polymerase concentrations on the test terrain, and the *R*2 was registered to 0.917. In addition to that, robustness analysis across weather conditions demonstrated the capability of the model to generalize its performance being tested across a variety of weather conditions, and specifically, skies that tend to produce a poor performance. The fact that it has fewer latency requirements and can work with moderate amounts of memory further demonstrates how it would be an ideal candidate to deploy on smart microgrids using real-time edge computing.

The combination of feature embedding, temporal learning, and computation efficiency makes it possible that the TCN-EF model is as practical to apply in energy-sensitive applications as it is accurate. On the whole, the proposed work complements the future development of AI-based energy management systems as it presents a scalable and robust forecasting approach.

Future directions can be to consider multi-source renewable inputs (e.g. wind, hybrid PV-storage), to consider attention mechanisms to increase interpretability, and also to consider extending to probabilistic forecasting cases to allow risk-aware energy dispatch.

# REFERENCES

1. Y. Zhang, Y. Zou, X. Li, and J. Wang, “Microgrid energy management strategy considering source-load forecast error,” International Journal of Electrical Power & Energy Systems **164**, 110372 (2025).
2. N. Benti, M. Chaka, and A. Semie, “Forecasting renewable energy generation with machine learning and deep learning: current advances and future prospects,” Sustainability **15**, 7087 (2023).
3. S. Raja, A. Ahmad, M. Alqaraghuli, and W. Gao, “A review on the application of machine learning techniques for renewable energy forecast- ing,” Renewable and Sustainable Energy Reviews **149**, 111367 (2021).
4. C. Maduabuchi, C. Nsude, C. Eneh, E. Eke, K. Okoli, E. Okpara, C. Idogho, B. Waya, and C. Harsito, “Renewable energy potential estimation using climatic-weather-forecasting machine learning algorithms,” Energies **16**, 1603 (2023).
5. V. Saini, R. Kumar, A. Al-Sumaiti, and E. Heydarian-Forushani, “Learning based short term wind speed forecasting models for smart grid applications: an extensive review and case study,” Electric Power Systems Research **222**, 109502 (2023).
6. S. Al-Selwi, M. Hassan, S. Abdulkadir, A. Muneer, E. Sumiea, A. Alqushaibi, and M. Ragab, “Rnn-lstm: From applications to modeling techniques and beyond–systematic review,” Journal of King Saud University - Computer and Information Sciences **36**, 102068 (2024).
7. M. Karunarathne, P. Sudasinghe, H. Weeratunge, and D. Herath, “Improving microgrid energy demand forecasting using convolutional neural networks,” Engineer: Journal of the Institution of Engineers, Sri Lanka **57**, 1–10 (2024).
8. Y. He, P. Huang, W. Hong, Q. Luo, L. Li, and K.-L. Tsui, “In-depth insights into the application of recurrent neural networks (rnns) in traffic prediction: a comprehensive review,” Algorithms **17**, 398 (2024).
9. P. Sudasinghe, D. Herath, I. Karunarathne, H. Weeratunge, and L. Jayasuriya, “Forecasting renewable energy for microgrids using machine learning,” Discover Applied Sciences **7**, 449 (2025).
10. S. Silwal, C. Mullican, Y. Chen, A. Ghosh, J. Dilliott, and J. Kleissl, “Open-source multi-year power generation, consumption, and storage data in a microgrid,” Journal of Renewable and Sustainable Energy **13**, 025301 (2021).