Technical Report: Hourly Temperature Forecasting for Solar Scheduling

# Abstract

This report presents the development of a temperature forecasting system aimed at predicting hourly temperature values up to 24 hours of inputted date for selected regions. The motivation stems from the increasing need for precise weather forecasts in solar energy planning and scheduling. The system has been built with a focus on accuracy, performance, and ease of use, specifically tailored for the districts of Pune, Kolhapur, and Satara.

# 1. Introduction

# 1.1 Background

This project is a part of a broader initiative in solar energy forecasting. Temperature is one of the critical parameters influencing solar power generation. Under the guidance of guide and Prof. Ashok Thorat , the focus was narrowed to temperature prediction, particularly at an hourly scale, to aid in the solar energy scheduling process.

# 1.2 Objective

The main objective is to build a predictive system that can:  
- Forecast temperature values for 24 consecutive hours for a user-selected date.  
- Provide an intuitive and responsive user interface.  
- Maintain efficient back-end performance for quick response and future scalability.

# 1.3 Scope and Limitations

The system currently works only for three districts: Pune, Kolhapur, and Satara. It supports temperature prediction up to 24 hours ahead of a given date. Although additional weather parameters are available in the dataset, only temperature and datetime have been used in this version to reduce memory load and processing time.

# 2. Dataset Details

# 2.1 Source and Coverage

The data has been collected from the Open-Meteo platform, which provides hourly historical weather records. The time-frame spans from January 1, 2024, to June 20, 2025, offering sufficient data for training and evaluation.

# 2.2 Dataset Characteristics

The dataset used for this project contains approximately **183,000 rows** of hourly weather records. These records span from **January 1, 2024**, to **June 20, 2025**, and cover three districts: Pune, Kolhapur, and Satara. Among the various columns available in the dataset, the two most crucial ones for this version of the model are **temperature (in °C)** and **datetime**. The temperature column serves both as the input (past values) and the prediction target (future values), while the datetime column provides the timestamp for each hourly observation.

Although the dataset includes additional parameters such as humidity, wind speed, rainfall, surface pressure, sea-level pressure, and cloud cover, these features have not been included in the current phase. This decision was made to optimize model performance and reduce memory and computation load, especially since the project is being developed and deployed on local infrastructure. However, these additional features may be incorporated in future versions to enhance the accuracy of the predictions.

# 3. Methodologies

# 3.1 Project Architecture

The entire workflow is built around the standard data science process:  
1. Data Acquisition  
2. Pre-processing  
3. Feature Engineering  
4. Model Training  
5. Evaluation  
6. Deployment

# 3.2 Feature Engineering

- Added lag columns for past 24 hours  
- Created future columns for 0 to 23 hours ahead  
- Removed unused features  
- Dropped rows with missing values

# 3.3 Model Design

The model developed for this project falls under the category of **supervised learning**, as it is trained using historical input-output pairs. Specifically, it addresses a **multi-output regression problem**, where the goal is to predict multiple target values—in this case, hourly temperatures for the next 24 hours—from a single input instance representing the previous 24-hour temperature sequence.

For this purpose, a **Random Forest Regressor** has been employed. Random Forest was chosen because its robustness and consistent performance on structured data. Random Forest is one of the best algorithm for good balance between bias and variance, it has relatively more hyperparameter compared to other models in sciket learn, and is generally resistant to overfitting, especially when trained on a diverse dataset. Its ensemble-based approach makes it stable and reliable, which is critical for weather-related predictions where precision and generalization both matter.

# 3.4 Technologies and Tools

- Languages: Python

- Frontend : HTML, CSS, Java script

- Libraries: pandas, scikit-learn, joblib

- Backend: Flask

- ORM: SQLAlchemy

- Database: SQLite

# 4. User Interface Overview

The web application is divided into three key pages:  
- Input Page: User selects location and date.  
- Prediction Page: Displays hourly forecasts.  
- Analysis Page: Comparative model performance (planned).

# 5. Results and Evaluation

The model's performance was evaluated using the following metrics:  
- Mean Absolute Error  
- Root Mean Square Error  
- R² Score  
(Exact values will be added after final training.)

# 8. Future Scope

- Improve model efficiency  
- Enhance backend error handling  
- Expand forecast window  
- Add support for more regions

# 7. Conclusion

This project successfully demonstrates a method to forecast temperature for up to 24 hours using historical data. The application is efficient, simple, and suitable for solar scheduling tasks.

# Appendix (To be Attached)

- System Architecture Diagram  
- Model Hyperparameter Grid  
- Output Screenshots  
- Model Comparison (planned)