System Specification Document

S.O.L.A.R. Project

January 27, 2026

Revision History

Version	Date	Author	Description
0.1	2025-12-17	Matteo	Initial draft
1.0	2026-01-25	Matteo	Approved release

Contents

1 Project Summary

1.1 Scope

The scope of this project is the design and development of an end-to-end data-driven system for short-term solar power generation forecasting. The system covers the full lifecycle of a machine learning application, including data ingestion, preprocessing, model training, validation, deployment, monitoring, and incremental updates. The solution is designed to simulate real-world operational conditions through synthetic sensors. The project further incorporates a slightly simpler ILSTM model proposed in this paper as a proof of concept for advanced forecasting techniques.

1.2 Objectives

The objectives of the project are the following:

- Design and implement a reliable regression-based system for short-term solar power forecasting.
- Develop an incremental learning pipeline capable of adapting to evolving data distributions.
- Implement monitoring mechanisms for data drift and performance degradation.
- Integrate governance features such as metric logging, model evaluation, and reproducibility support.
- Provide a proof of concept for extending the system with an Incremental Long Short-Term Memory (ILSTM) model.

1.3 Relevance

This project is relevant in the context of renewable energy management, where accurate short-term forecasting plays a critical role in ensuring grid stability, optimizing energy dispatch, and reducing imbalance penalties. Reliable predictions of solar power generation enable energy operators to better align supply with demand and to improve the overall efficiency of power system operations.

The project demonstrates the practical application of machine learning and neural network techniques for time-series forecasting, focusing on the prediction of next-day solar power generation using historical production data and weather-related features. By addressing real-world challenges such as data variability and evolving system behavior, the project highlights the value of data driven solutions in supporting sustainable energy management.

2 Deliverables

- A data ingestion and preprocessing pipeline supporting historical data.
- An incremental learning regression model for next-timestamp solar power forecasting.
- A server-side prediction service implemented in Flask, exposing a REST-style API.
- An interactive Streamlit dashboard for visualization of predictions and monitoring metrics.
- A proof-of-concept implementation and evaluation of an incremental LSTM (ILSTM) model.
- A DAO interface for querying sensor data, including support for synthetic sensor sources.
- Technical documentation describing system architecture, model design, and monitoring strategy.

2.1 Milestones

1. Completion of dataset analysis and validation of data quality.
2. Development and evaluation of a baseline incremental regression model.
3. Implementation of the DAO layer and integration of synthetic sensor data sources.
4. Deployment of the Flask-based prediction service exposing a REST-style API.
5. Development of the Streamlit dashboard for real-time visualization and monitoring.
6. Activation of monitoring components for data drift and prediction performance.
7. Implementation and evaluation of an advanced incremental LSTM (ILSTM) model.

2.2 Work Breakdown Structure

The project started by a brainstorming session to identify the main components and tasks required to achieve the project objectives. The WBS was then structured into the following main components:

2.2.1 Project Manager

- Coordination of development activities and task scheduling.
- Redacting and maintaining project documentation.
- Review and validation of technical documentation and deliverables.

2.2.2 Data Analyst / ML Engineer

- Exploratory data analysis and validation of the datasets.
- Development and evaluation of the baseline incremental regression model.
- Integration of monitoring components for drift detection and performance tracking.
- Design, implementation, and evaluation of the incremental LSTM (ILSTM) model.
- Definition and monitoring of performance metrics and KPIs.

2.2.3 Software Developer

- Implementation of the DAO layer for accessing synthetic sensor data.
- Development of the Flask-based prediction service and REST-style API.
- Development of the Streamlit dashboard for visualization and monitoring.

2.3 Sprint Plan

Due to the vacation period, the length of the sprints has been adjusted to accommodate the team members' availability. The sprint plan is as follows:

Sprint	Start Date	Activities
Sprint 0	–	Brainstorming and definition of project requirements, along with analysis of the historical dataset.
Sprint 1	19-12-2025	Implementation of the core system backbone, including the Streamlit dashboard, Flask service, and DAO layer; preprocessing of historical data and training of the first incremental regression model.
Sprint 2	23-12-2025	API refinement, integration of the synthetic sensor simulation, and delivery of the first end-to-end basic model.
Sprint 3	16-01-2026	Development of the Minimum Viable Product (MVP), integration of ADWIN-based drift detection, and implementation of the incremental LSTM (ILSTM) model.
Sprint 4	23-01-2026	Final model tuning, full system deployment, and completion of monitoring and documentation.

2.4 Definition of Ready (DoR)

A task or feature is considered ready to start when all the following conditions are met:

- The task is clearly defined, with measurable objectives and scope.
- Required datasets are available, cleaned, and preprocessed for use with synthetic sensors.
- Technical specifications, including API endpoints, dashboard requirements, and model input/output, are documented.
- Dependencies, such as necessary libraries (Flask, Streamlit, River), frameworks, or environment setup, are identified and accessible.
- Acceptance criteria are clearly defined, including performance targets for the incremental regression model and drift detection.

2.5 Definition of Done (DoD)

A task or feature is considered done when all the following conditions are satisfied:

- The model is implemented, integrated into the Flask API, and correctly returns predictions to the dashboard.
- The synthetic sensor interface is functional, providing real-time simulated data for predictions.
- Incremental learning updates the model continuously, and ADWIN drift detection monitors feature distributions and triggers alerts appropriately.
- Model performance meets the predefined KPIs, such as daily mean MSE below the target threshold.
- The codebase is fully documented, versioned, and reviewed according to project standards.
- Basic functional tests of the entire pipeline (API, dashboard, monitoring, and sensor interface) have passed successfully.
- All related documentation, including design decisions and system architecture, is completed and available.

2.6 Resources and Infrastructure

- **Programming Languages and Libraries:**
  - Python: Core language for data processing, modeling, and backend development.
  - Libraries: pandas, numpy, scikit-learn, river (for real-time machine learning), Flask (for REST API), Streamlit (for dashboard visualization), TensorFlow and Keras (for ILSTM model implementation).
- **Data Management:**
  - Raw datasets from solar power plants, including sensor data for temperature, irradiation, and power generation are stored as immutable objects.
  - Clean CSV files for model training are generating through deterministic preprocessing.
  - At runtime DAOs retrive data for the server.
- **Version Control:**
  - GitHub: Used for version control, collaboration, and documentation.
- **Development Tools:**
  - Jupyter Notebooks: For exploratory data analysis and prototyping.
  - VS Code: Integrated development environment for coding and debugging.
- **Infrastructure:**
  - Local development environment for testing and debugging.
- **Collaboration Tools:**
  - Documentation: LaTeX for project documentation and reporting.