Solar Panel Output Prediction Using ML With Weather Data

Yovel Jirad & Shay Pinsky

Braude College Of Engineering

Capstone Project Phase A - 61998

Supervisors: Dr. Dan Lemberg and Mrs. Elena Kramer

April 2025

Contents

1 Abstract

Solar energy forecasting represents a critical challenge for the effective integration of photovoltaic systems into modern electrical grids. The inherent variability of solar power generation due to changing meteorological conditions creates significant difficulties for grid operators who must maintain real-time balance between energy supply and demand while ensuring system stability.

This project develops an advanced machine learning system for solar energy output prediction using comprehensive weather data analysis. The system implements a dual-architecture approach combining Gated Recurrent Unit (GRU) networks for short-term forecasting and Autoformer transformer variants for long-term prediction. The GRU model generates hourly predictions for the next 24 hours using recursive forecasting methodology, while the Autoformer architecture produces daily predictions for 2-4 days ahead through direct multi-step forecasting with autocorrelation mechanisms designed specifically for capturing seasonal and cyclical patterns in meteorological data.

Weather data is collected in real-time from the Israel Meteorological Service (IMS) API, including temperature, solar radiation, humidity, and atmospheric pressure measurements. Due to unavailability of direct solar farm data, the system employs theoretical photovoltaic output calculations based on established research methodologies to generate training labels for machine learning models. Feature selection utilizes Pearson correlation analysis to identify the most relevant meteorological parameters for accurate solar energy prediction.

The system targets prediction accuracy within a 10% error margin for day-ahead forecasts, validated through comprehensive evaluation metrics including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). The forecasting capabilities are delivered through a web-based interface that displays real-time predictions, historical accuracy tracking, and interactive visualizations for both operational grid management and strategic energy planning applications.

This dual-model approach addresses the fundamental challenge that different forecasting horizons require specialized architectures, leveraging computational efficiency of GRU networks for immediate operational needs while utilizing Autoformer's advanced temporal modeling capabilities for strategic planning requirements. The system contributes to renewable energy integration by providing reliable solar energy forecasts that support optimal grid operations and infrastructure planning decisions.

2 Introduction

Solar energy has emerged as one of the most promising renewable energy sources for addressing global climate change and reducing dependence on fossil fuels. As photovoltaic (PV) technology becomes increasingly cost-effective and widespread, the integration of solar power into electrical grids presents both opportunities and challenges. The inherent variability of solar energy generation, driven by changing weather conditions, creates significant difficulties for grid operators who must balance energy supply and demand in real-time while maintaining grid stability and reliability.

The transition toward renewable energy systems has accelerated globally, with solar installations experiencing exponential growth over the past decade. However, this rapid expansion has highlighted a critical challenge: the unpredictable nature of solar energy production due to meteorological variability. Unlike conventional power generation systems that can be controlled and dispatched on demand, solar energy output fluctuates based on atmospheric conditions, seasonal patterns, and daily cycles. This variability introduces complexity into energy market operations, grid management, and long-term energy planning processes.

This project is categorized as a machine learning development & research project as it involves building a predictive model that uses historical weather data (temperature, radiation, humidity, etc.) to forecast solar energy production in specific locations at set time frames. The development process requires creating a pipeline for data processing, selecting appropriate ML algorithms, training models on the weather-to-energy-output relationship, and optimizing for accurate predictions that could inform solar energy infrastructure planning and operations.

Traditional solar energy forecasting methods often rely on physical models that simulate atmospheric processes through numerical weather prediction systems. While these approaches provide valuable insights into atmospheric dynamics, they face significant limitations in handling non-linear relationships between multiple meteorological parameters and solar output. Physical models also struggle with computational efficiency when processing large datasets and often require extensive calibration for specific geographical locations and installation configurations.

Machine learning models offer significant advantages over traditional approaches by effectively capturing complex, non-linear patterns between weather parameters and solar output without requiring explicit mathematical formulation of physical relationships. These data-driven approaches can automatically identify and learn intricate dependencies within historical datasets, making them particularly well-suited for solar energy forecasting applications where multiple meteorological factors interact in complex ways.

For this project, we have strategically selected different neural network architectures optimized for specific forecasting horizons. For short-term predictions, we will employ Gated Recurrent Unit (GRU) networks, which excel at capturing immediate temporal dependencies while maintaining computational efficiency. GRUs use fewer parameters than traditional LSTM networks, making them faster to train and less prone to overfitting while still effectively addressing the vanishing gradient problem that affects simple recurrent neural networks. This makes them ideal for real-time and next-day solar energy forecasting where rapid processing and high accuracy are essential.

For long-term predictions extending several days into the future, we will implement the Autoformer architecture, a specialized transformer variant designed specifically for extended time series forecasting. Autoformer replaces standard attention mechanisms with autocorrelation mechanisms that measure time-delay similarity between input signals, making it particularly effective for solar energy data that exhibits strong seasonal and daily patterns. The autocorrelation approach allows the model to identify and leverage recurring patterns in weather data that influence solar generation over longer time horizons.

The motivation for this dual-architecture approach stems from the recognition that different forecasting horizons require different modeling strategies. Short-term forecasting benefits from models that can quickly process recent weather patterns and generate immediate predictions, while long-term forecasting requires architectures capable of identifying and extrapolating seasonal trends and cyclic patterns that emerge over extended periods.

This project addresses the critical need for reliable solar energy forecasting to support the growing integration of renewable energy sources into power systems. Accurate predictions enable better energy management by allowing grid operators to anticipate solar generation fluctuations and adjust conventional generation accordingly. This capability reduces operational costs through improved unit commitment and economic dispatch decisions, while also enhancing grid stability by providing advance warning of potential supply variations.

Furthermore, reliable solar forecasting supports optimal planning of solar installations by enabling developers and investors to assess the economic viability of proposed projects based on expected energy yields. This capability is particularly valuable for identifying optimal locations for new solar infrastructure by analyzing historical weather patterns across regions to maximize potential energy returns and return on investment.

By developing a comprehensive prediction system that leverages state-of-the-art machine learning techniques specifically selected for different forecasting horizons, this project contributes to the advancement of renewable energy forecasting methodologies. The dual-model approach combining GRU networks for short-term precision and Autoformer for long-term reliability represents an innovative solution to the multi-horizon forecasting challenge that has significant implications for supporting the global transition toward sustainable energy systems.

The successful implementation of this forecasting system will provide a valuable tool for energy planners, grid operators, and solar energy developers, ultimately supporting more effective integration of solar power into modern electrical grids and contributing to the broader goals of renewable energy adoption and climate change mitigation.

3 Problem Statement

The challenge of accurately predicting solar energy production represents a significant barrier to the widespread adoption and effective integration of photovoltaic systems into modern electrical grids. Solar energy generation is inherently fluctuating due to changing atmospheric conditions, making it difficult to reliably integrate into power grids that require precise balance between energy supply and demand.

Current forecasting methods face several critical limitations that this project aims to address:

Multi-horizon Prediction Challenges: Solar energy planning requires reliable predictions across multiple time horizons, from real-time adjustments to long-term infrastructure planning. Most current systems are optimized for either short-term or long-term forecasting, lacking the versatility to provide accurate predictions across different time scales within a unified framework.

Weather Parameter Integration: While solar radiation is the primary driver of photovoltaic output, other meteorological factors such as temperature, humidity, atmospheric pressure, and cloud cover significantly influence solar panel efficiency. Many existing systems fail to effectively integrate and weigh these multiple weather parameters in their prediction models, limiting their accuracy and reliability.

Geographic and Temporal Specificity: Solar energy prediction models must account for location-specific weather patterns and seasonal variations. Generic models often fail to capture the unique climatic characteristics of specific regions, reducing their effectiveness for local solar energy planning and operations.

Real-time Data Processing: Modern solar energy management requires the ability to process real-time weather data and generate immediate predictions. Many existing systems lack the computational efficiency and data processing capabilities needed for real-time forecasting applications.

This project addresses these challenges by developing an advanced machine learning system that can accurately predict solar energy output across multiple time horizons using comprehensive weather data analysis. The system aims to achieve prediction accuracy within a 10% error margin for day-ahead forecasts while providing reliable predictions for up to four days in advance, supporting both operational grid management and strategic energy planning decisions.

4 Literature Review

4.1 Solar Energy Forecasting: Fundamentals and Applications

4.1.1 Photovoltaic Output Dependencies

Solar energy generation is fundamentally dependent on meteorological conditions, with solar radiation serving as the primary driving factor. Jebli et al. [1] established the quantitative relationship between weather parameters and photovoltaic output through the equation:

$$P_{out,pv} = C_{R,PV} \frac{G_T}{G_{T,STC}} [1 + \alpha_p (T_C - T_{C,STC})]$$

where $P_{out,pv}$ represents the predicted photovoltaic power output in watts, $C_{R,PV}$ is the rated capacity of the photovoltaic array under standard test conditions, G_T denotes the actual solar radiation incident on the photovoltaic panel surface measured in W/m^2, and $G_{T,STC}$ represents the solar radiation under standard test conditions (typically 1000 W/m^2). The temperature-related variables include T_C, which is the current photovoltaic cell temperature in degrees Celsius, $T_{C,STC}$ representing the cell temperature under standard test conditions (25°C), and α_p indicating the temperature coefficient of power expressed as a percentage per degree Celsius, which accounts for the efficiency loss of solar panels as temperature increases above the standard conditions.

The study emphasized that using all available meteorological features can lead to overfitting and reduced model performance, highlighting the importance of systematic feature selection based on correlation analysis. Wind speed and direction were found to be less relevant for solar energy prediction, making temperature, solar radiation, humidity, and pressure the optimal parameter set for forecasting models.

4.1.2 Forecasting Horizons and Operational Requirements

Solar energy forecasting serves different operational needs across multiple time horizons. Real-time forecasting supports immediate grid balancing and storage control decisions, while short-term predictions (hours to next day) are essential for energy trading and unit commitment planning. Medium-term forecasting (2-4 days ahead) enables maintenance scheduling and grid stability planning, while long-term predictions support infrastructure development and investment decisions.

Each forecasting horizon presents unique challenges in terms of accuracy requirements and acceptable error margins. Jebli et al. [1] demonstrated that different model architectures excel at different horizons, with their research showing that unified approaches using single models for all horizons often compromise performance compared to specialized multi-model strategies.

4.2 Time Series Forecasting Methodologies

4.2.1 Fundamental Approaches

Time series forecasting for solar energy involves capturing temporal dependencies in weather data to predict future photovoltaic output. The temporal nature of meteorological data introduces several key characteristics that must be addressed: trend components representing long-term changes, seasonal patterns reflecting daily and annual cycles, and stochastic variations due to weather unpredictability.

Traditional approaches have relied on statistical methods such as autoregressive models (ARIMA) and physical models based on numerical weather prediction systems. However, these methods face limitations in capturing complex non-linear relationships between multiple meteorological parameters and solar output, particularly when dealing with high-dimensional weather datasets.

4.2.2 Forecasting Strategies: Recursive vs Direct Approaches

Two primary strategies exist for multi-step time series forecasting: recursive forecasting and direct multi-step (one-shot) forecasting. Recursive forecasting predicts one time step ahead and uses that prediction as input for subsequent steps, creating an autoregressive loop. This approach offers high accuracy for short-term predictions but suffers from error propagation over longer horizons as small errors accumulate and magnify.

Direct multi-step forecasting predicts multiple time steps simultaneously in a single operation, eliminating error propagation issues. This approach proves more suitable for medium to long-term forecasting where capturing broader trends is more important than fine-grained precision at individual time steps. The choice between these strategies significantly impacts model architecture selection and performance across different forecasting horizons.

4.3 Deep Learning Architectures for Solar Forecasting

4.3.1 Recurrent Neural Networks and Variants

Recurrent Neural Networks represent a fundamental architecture for sequential data processing in time series forecasting. Traditional RNNs maintain hidden states that carry information across time steps, enabling them to learn temporal dependencies. However, basic RNNs suffer from vanishing gradient problems that limit their ability to capture long-term dependencies effectively.

Long Short-Term Memory (LSTM) networks address these limitations through sophisticated gating mechanisms that control information flow. LSTMs use input, forget, and output gates to selectively retain and update information, making them effective for capturing both short and long-term temporal patterns in solar energy data. Casolaro et al. [2] demonstrate that LSTMs consistently outperform traditional forecasting methods in solar applications through their ability to model complex relationships between historical weather patterns and photovoltaic output.

4.3.2 Hyperparameter Optimization in Neural Networks for Time Series

Hyperparameters are configuration settings that control the learning process and architecture of machine learning models, distinct from the model parameters (weights and biases) that are learned during training. These settings must be specified before training begins and significantly influence model performance, convergence speed, and generalization capability.

Effective solar energy forecasting requires systematic optimization of model hyperparameters to achieve optimal performance. Jebli et al. [1] demonstrated that model configuration significantly impacts prediction accuracy, with their optimized models achieving R^2 values of 92.61% for real-time and 93% for daily predictions through careful parameter selection.

For recurrent networks, critical hyperparameters include sequence length (input window size), hidden units, learning rates, and regularization parameters. Casolaro et al. [2] emphasize that GRU networks benefit from systematic tuning of dropout rates (typically 0.1-0.3) and hidden unit counts (32-128) to balance model complexity with overfitting prevention.

Transformer-based architectures require attention to embedding dimensions, number of attention heads, and layer depths. The Autoformer architecture particularly benefits from optimization of autocorrelation parameters and decomposition window sizes for effective seasonal pattern recognition in solar energy data.

The selection of appropriate hyperparameters becomes particularly crucial in time series forecasting, where temporal dependencies and seasonal patterns must be captured effectively while avoiding overfitting to historical weather patterns that may not generalize to future conditions.

4.3.3 Gated Recurrent Units for Short-Term Forecasting

Gated Recurrent Units (GRUs) represent an evolution of LSTM networks, offering similar capabilities with reduced computational complexity. GRUs employ two gating mechanisms - update and reset gates - compared to LSTM's three gates, resulting in fewer parameters and faster training while maintaining effectiveness in handling sequential dependencies.

Research demonstrates that GRUs provide advantages for short-term forecasting applications due to their computational efficiency and reduced overfitting tendency. The simplified architecture makes GRUs particularly suitable for real-time solar energy prediction where processing speed and immediate response are critical operational requirements. Studies show that GRUs can achieve comparable accuracy to LSTMs while requiring significantly less computational resources, making them optimal for next-day solar forecasting applications.

Our selection of GRU networks for short-term forecasting was informed by comprehensive analysis of recent comparative studies [4], which demonstrated GRU's superior performance in multivariate time series applications while maintaining computational efficiency essential for real-time solar energy prediction requirements.

4.3.4 Transformer Architectures for Long-Term Forecasting

Transformer architectures have revolutionized sequence modeling through attention mechanisms that process entire sequences simultaneously rather than sequentially. The standard transformer uses scaled dot-product attention computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

However, standard transformers face limitations in time series applications due to their locality-agnostic nature and memory bottleneck issues with long sequences.

4.3.5 Autoformer for Extended Solar Energy Prediction

Autoformer represents a specialized transformer variant designed specifically for long-term time series forecasting. Unlike standard transformers, Autoformer replaces attention mechanisms with autocorrelation mechanisms that measure time-delay similarity between input signals. This approach proves particularly effective for solar energy data that exhibits strong seasonal and daily patterns.

The autocorrelation mechanism aggregates the top-k similar sub-series to produce outputs, making it well-suited for identifying and leveraging recurring patterns in weather data over extended periods. Casolaro et al. [2] report that Autoformer achieves 38% relative improvement over traditional transformer approaches in energy forecasting applications, with superior performance in capturing long-term dependencies essential for multi-day solar energy prediction.

Additionally, Autoformer incorporates a decomposition module that breaks down time series into trend and seasonal components, assuming additive relationships. This decomposition approach aligns well with solar energy data characteristics, where seasonal variations and long-term trends significantly influence generation patterns.

The decision to implement Autoformer for long-term forecasting was based on extensive evaluation of transformer variants documented in recent literature [2, 6, 5, 7, 8], which consistently demonstrated Autoformer's superior capability in capturing seasonal patterns and long-term dependencies essential for multi-day solar energy prediction.

4.4 Multi-Horizon Forecasting Strategies

4.4.1 Dual-Architecture Approaches

Recent research emphasizes the advantages of specialized architectures for different forecasting horizons rather than attempting to optimize single models for all time scales. Jebli et al. [1] demonstrated that models combining different neural network architectures for specific horizons significantly outperform single-architecture approaches across comprehensive evaluation metrics.

The rationale for dual-architecture strategies stems from the fundamental differences in temporal pattern requirements across forecasting horizons. Short-term predictions benefit from models optimized for immediate temporal dependencies and rapid processing, while long-term forecasting requires architectures capable of identifying and extrapolating seasonal trends and cyclic patterns over extended periods.

This architectural selection strategy was further validated through analysis of current transformer implementations [1, 3], which provided practical insights into the computational trade-offs and performance characteristics that guided our final dual-model approach combining GRU efficiency with Autoformer's long-term modeling capabilities.

4.4.2 Architecture Selection Rationale

The selection of GRU networks for short-term forecasting and Autoformer for long-term prediction represents an optimal combination based on established research findings. GRUs provide the computational efficiency and immediate pattern recognition needed for next-day predictions, while Autoformer's autocorrelation mechanisms excel at identifying the recurring seasonal and daily patterns essential for multi-day forecasting.

This dual approach addresses the specific challenges of solar energy forecasting where immediate accuracy for grid operations must be balanced with reliable medium-term predictions for planning purposes. The combination leverages the strengths of each architecture while avoiding the compromises inherent in unified approaches.

4.5 Parameter Selection and Correlation Analysis

4.5.1 Pearson Correlation for Meteorological Parameter Selection

Effective solar energy forecasting requires systematic identification of the most relevant meteorological inputs. Jebli et al. [1] employed Pearson correlation coefficient analysis to establish quantitative relationships between weather parameters and solar output. Their methodology demonstrated that solar radiation and temperature show the strongest positive correlations (approaching 1.0), while humidity exhibits significant negative correlation (-0.36).

The research established that atmospheric pressure shows weaker correlation (0.12) but remains relevant due to its relationship with other meteorological variables. Wind parameters (speed and direction) were found to have minimal direct impact on solar energy generation, though wind direction showed moderate correlation with temperature (0.33) and pressure (0.42).

4.5.2 Data Quality and Preprocessing Requirements

Solar energy forecasting models require comprehensive data preprocessing to handle missing values, normalize measurements across different parameter scales, and ensure consistent temporal resolution. The integration of real-time meteorological data from services like the Israel Meteorological Service (IMS) introduces additional challenges related to data latency, quality control, and synchronization across multiple weather stations.

Effective preprocessing strategies must address temporal gaps in weather measurements, particularly for solar radiation data that may be affected by equipment maintenance or adverse weather conditions. Interpolation techniques appropriate for time series data become critical for maintaining model performance when dealing with incomplete meteorological datasets.

4.6 Performance Evaluation and Benchmarking

4.6.1 Evaluation Metrics for Solar Forecasting

Solar energy forecasting model evaluation requires comprehensive metrics addressing both accuracy and practical applicability. Standard metrics include Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and normalized variants that enable comparison across different scales and installations. Jebli et al. [1] demonstrated the importance of using multiple evaluation metrics, achieving R-squared values of 92.61% for real-time predictions and 93% for daily forecasts.

The research emphasized that model evaluation must consider multiple forecasting horizons and real-world deployment scenarios. Normalized Root Mean Square Error (NRMSE) provides particularly valuable insights for solar applications by enabling comparison across different geographic locations and seasonal conditions.

4.6.2 Benchmark Performance and Expectations

Established research provides performance benchmarks for solar energy forecasting across different horizons and methodologies. Jebli et al. [1] achieved significant improvements over traditional approaches, with their optimized models demonstrating superior performance compared to linear regression and support vector regression methods.

The generalizability of forecasting approaches has been validated through cross-regional testing, with consistent performance demonstrated across different climate conditions from semi-desert regions to tropical environments. This research provides important precedent for developing robust, transferable solar energy forecasting systems applicable to diverse geographic and climatic conditions.

4.7 Research Gaps and Project Positioning

4.7.1 Current Limitations in Multi-Horizon Forecasting

Existing solar energy forecasting research predominantly focuses on single-architecture solutions optimized for specific forecasting horizons. Limited research addresses the development of integrated systems that combine specialized architectures for different time scales within unified forecasting platforms. Additionally, most studies rely on synthetic or limited real-world datasets, with insufficient validation using comprehensive meteorological data from operational weather services.

4.7.2 Contribution of Current Project

This project addresses identified research gaps through several key contributions: implementation of a dual-architecture approach combining GRU networks for short-term accuracy with Autoformer for long-term reliability, integration with real-time meteorological data from the Israel Meteorological Service for operational validation, and development of a comprehensive forecasting system that addresses both immediate operational needs and strategic planning requirements.

The project is implemented in Israeli climatic conditions, providing valuable validation of forecasting methodologies in semi-arid environments, contributing to the geographic diversity of solar energy forecasting research and demonstrating practical applicability for Mediterranean and Middle Eastern solar installations. However, training the same model with weather data of any other location should make a location specific model for that area.

5 Expected Achievements

5.1 System Goals

The primary goal of this project is to develop a solar energy forecasting website that showcases our predictions alongside comprehensive statistics. The website will serve as both an informative platform for general users and a detailed resource for evaluating the accuracy of our forecasting models.

At its core, the website will rely on our advanced machine learning models to deliver accurate, real-time solar energy output forecasts based on weather parameters. The backend will be powered by our forecasting system, which will automatically import real-time weather data from meteorological services. This data will be processed by our models to generate reliable predictions for upcoming days.

In addition to providing real-time forecasts, the website will include:

- Current Weather Data: Displaying real-time weather parameters, including temperature, humidity, wind speed, and solar radiation.
- Interactive Comparison Tool: A feature that allows users to compare our model's forecasts with traditional prediction methods, offering transparency in the accuracy of machine learning-based forecasting versus conventional approaches.
- Research and Development Section: Users will also have the option to read about and explore our project methodology, including the architecture we developed, the implementation of the models, and other technical insights.

5.2 Prediction Accuracy Expectations

Our solar energy output prediction system will focus on achieving:

- Error margin of no more than 10% for day-ahead forecasts
- Reliable predictions for up to four days ahead
- Performance matching or exceeding traditional forecasting methods

To accomplish these goals, we will utilize a dual-architecture forecasting approach:

- Model 1: Provides hourly predictions for the next 24 hours using a recursive method
- Model 2: Generates predictions for subsequent days using a one-shot approach

By utilizing specialized models for different horizons, we aim to achieve optimal accuracy across both short-term and long-term forecasting periods.

6 Engineering Process

6.1 Progress and Results So Far

Data Collection Strategy and Implementation

Our data collection strategy began with the establishment of reliable sources for both weather and solar energy data. We contacted the Israel Meteorological Service (IMS) through their official website to obtain API access credentials for their comprehensive weather database. After reviewing their available meteorological parameters, we determined that IMS data would provide an excellent foundation for our project.

Through the IMS API, we successfully obtained historical and real-time weather data from multiple monitoring stations across Israel. The collected parameters include temperature, humidity, wind speed, wind direction, atmospheric pressure, and solar radiation measurements - all critical factors that directly influence solar panel energy output according to our literature review.

Solar Energy Data Acquisition Challenges

For solar energy output data, we initially planned to use the free access data from the www.noga-iso.co.il website, Noga being the company for electrical grid management working in association with the Israel Electric Corporation (Hevrat Hashmal). However, the data proved insufficient for our modeling requirements for two key reasons: first, it lacked essential details such as the number of solar panels in each installation; second, it was aggregated at the national level rather than providing location-specific measurements that would correspond to our weather station data.

Having identified the limitations with the free access data, we approached the company Noga through official channels, seeking more detailed solar installation data. Unfortunately, the requested information belongs to private corporations and Noga cannot share it legally. We then found several private companies and tried to contact them. However, after careful consideration, they declined to share such sensitive commercial data for research purposes.

Theoretical Approach Adoption

Due to these data acquisition constraints, we adopted the theoretical approach established in the literature. Based on the methodologies presented in the papers by Jebli et al. [1], we decided to implement the photovoltaic output calculation equation to generate accurate solar energy labels for our machine learning models using the weather parameters as inputs.

Exploratory Data Analysis and Pattern Recognition

Our exploratory data analysis commenced with the systematic examination of meteorological data obtained through the IMS API from multiple weather monitoring stations distributed across Israel. This comprehensive investigation aimed to understand the fundamental characteristics and patterns within the weather parameters that directly influence solar energy generation.

The analysis concentrated on meteorological variables with established correlations to photovoltaic output, primarily focusing on solar radiation, ambient temperature, relative humidity, and atmospheric pressure. These parameters were selected based on their demonstrated significance in solar energy prediction literature, particularly following the findings of Jebli et al. [1], who identified these as the most influential factors through Pearson correlation analysis.

Photovoltaic Output Calculation Implementation

A critical component of our exploratory phase involved implementing the photovoltaic output calculation methodology described in the research literature. Based on the equation presented in Jebli et al. [1], we utilized the relationship between solar radiation and temperature to generate synthetic solar energy output labels:

$P_{out,pv} = C_{R,PV} \frac{G_T}{G_{T,STC}} [1 + \alpha_p (T_C - T_{C,STC})]$

where $P_{out,pv}$ represents the predicted photovoltaic power output in watts, $C_{R,PV}$ is the rated capacity of the photovoltaic array under standard test conditions, G_T denotes the actual solar radiation incident on the photovoltaic panel surface measured in W/m^2, and $G_{T,STC}$ represents the solar radiation under standard test conditions (typically 1000 W/m^2). The temperature-related variables include T_C, which is the current photovoltaic cell temperature in degrees Celsius, $T_{C,STC}$ representing the cell temperature under standard test conditions (25°C), and α_p indicating the temperature coefficient of power expressed as a percentage per degree Celsius, which accounts for the efficiency loss of solar panels as temperature increases above the standard conditions.

This theoretical approach allowed us to create reliable ground truth labels for our machine learning models in the absence of direct solar farm data, establishing a scientifically validated relationship between meteorological measurements and expected solar energy output.

Data Quality Assessment and Preprocessing Preparation

During the data examination process, we encountered various data quality challenges that required systematic resolution. Missing values in the meteorological records were identified and addressed through planned interpolation techniques appropriate for time series data. We plan to implement normalization procedures to standardize measurements across different parameter scales, ensuring consistent input ranges for our machine learning models.

The exploratory analysis revealed distinct temporal patterns in the meteorological data, including daily temperature cycles, seasonal radiation variations, and humidity fluctuations that correlate with solar panel efficiency.

Through this thorough examination of the characteristics of the weather data and their relationships with solar energy generation, we established the foundation for developing robust predictive models tailored specifically to the climatic conditions and solar potential of the region.

Model Research and Architecture Selection

Our preliminary model testing phase focused on comprehensive research and analysis of advanced machine learning architectures specifically designed for time series forecasting applications. Rather than implementing these models immediately, we conducted extensive literature review and technical evaluation to identify the most suitable approaches for solar energy prediction based on meteorological data.

Short-Term Forecasting Architecture Selection

Through our research analysis, we determined that a dual-architecture approach would be optimal for our solar energy forecasting requirements. For short-term predictions, we identified Gated Recurrent Unit (GRU) networks as the most appropriate choice based on their demonstrated effectiveness in capturing temporal dependencies while maintaining computational efficiency. As documented by Casolaro et al. [2], GRU networks are computationally efficient while effectively handling sequential data while using fewer parameters than LSTM, making it faster to train and less prone to overfitting, while still effectively addressing the vanishing gradient problem that plagues simple recurrent neural networks.

Long-Term Forecasting Architecture Selection
For long-term solar energy forecasting, our research led us to select the Autoformer architecture, a novel transformer variant specifically designed for extended time series prediction. According to Casolaro et al. [2], the Autoformer replaces standard scaled dot-product attention with an autocorrelation mechanism and employs a decomposition module to break down time series into trend and seasonal components, making it particularly well-suited for solar energy data that exhibits strong seasonal and daily patterns. Our supplementary web research revealed that Autoformer demonstrates state-of-the-art performance in long-term forecasting with a 38% relative improvement over traditional transformer approaches across multiple benchmarks including energy applications.

Literature-Based Validation
The foundational methodology for our approach is supported by Jebli et al. [1], who demonstrated that Pearson correlation coefficient can effectively identify the most relevant meteorological inputs for solar energy prediction models. Their work established that photovoltaic output depends on meteorological impact factors including solar radiation, temperature, humidity, wind speed, and pressure, providing the theoretical basis for our feature selection approach.

Our selection of these architectures was further validated through extensive web research, which confirmed their proven effectiveness in energy forecasting applications. Studies specifically demonstrate Autoformer's superior performance in photovoltaic power prediction tasks, with significant improvements in forecasting accuracy for solar energy applications. The combination of GRU for short-term precision and Autoformer for long-term reliability aligns perfectly with our dual-model forecasting strategy, where each architecture can leverage its specific strengths for different prediction horizons.

This research-driven approach to model selection, supported by both peer-reviewed literature and current web-based technical resources, ensures that our final implementation will be built upon architectures with proven track records in both time series forecasting generally and solar energy prediction.

6.2 Suggested Architecture & Model

6.2.1 Dual-Model Approach and System Design

Dual-Model Approach Rationale
Our solar energy forecasting system implements a dual-architecture approach that combines specialized neural networks optimized for different forecasting horizons. This design addresses the fundamental challenge that no single model architecture can effectively handle both immediate operational needs and strategic planning requirements across multiple time scales.

The proposed architecture integrates two complementary forecasting models within a unified prediction pipeline:

1. **Short-Term Model (GRU):** Generates hourly predictions for the next 24 hours using recursive forecasting methodology

2. **Long-Term Model (Autoformer):** Produces daily predictions for 2-4 days ahead using direct multi-step forecasting approach

System Integration and Data Flow

The system processes meteorological data through a structured pipeline designed to optimize prediction accuracy across different time horizons:

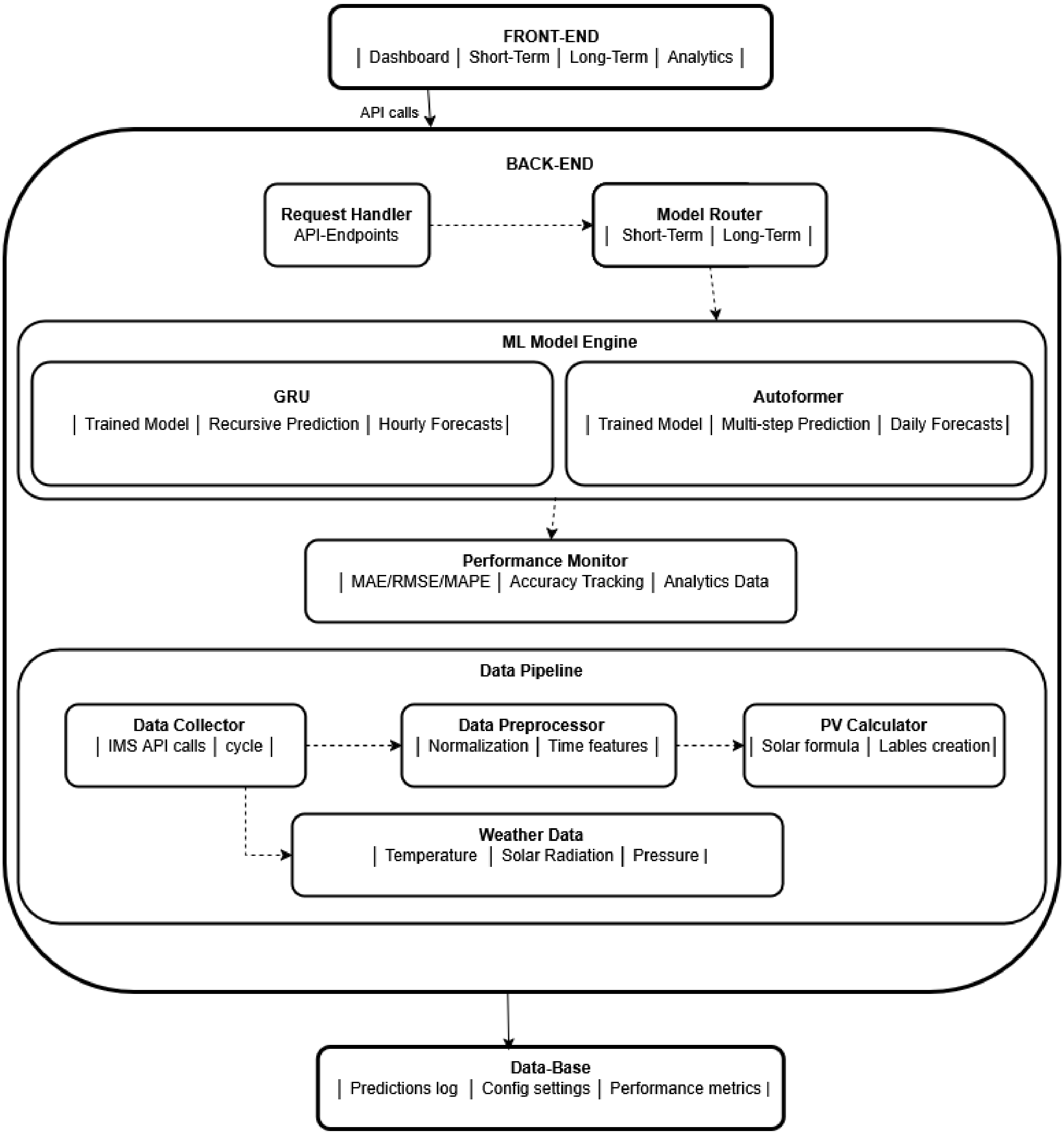

Figure 1: Overall System Architecture and Data Flow

Component Integration Strategy

Data Acquisition Layer:

- Real-time weather data ingestion from Israel Meteorological Service (IMS) API

- Data quality validation and missing value detection
- Automatic data refresh cycles every 30 minutes

Data Preprocessing Module:

- Pearson correlation-based parameter selection for optimal meteorological inputs
- Temporal data encoding (cyclic time representations)
- Data normalization and scaling for consistent model inputs

Dual-Model Prediction Engine:
The dual-model approach operates through coordinated prediction generation:

1. **Intelligent Routing:** System routes requests to appropriate model based on forecast horizon
2. **Temporal Partitioning:** GRU handles 0-24 hour forecasts, Autoformer manages 1-4 day predictions
3. **Independent Processing:** Each model operates independently for optimal performance

Solar Energy Calculation Integration

Our system incorporates the theoretical photovoltaic output calculation to generate training labels:

$$P_{out,pv} = C_{R,PV} \frac{G_T}{G_{T,STC}} [1 + \alpha_p (T_C - T_{C,STC})]$$

Where the system applies standard photovoltaic parameters:

- $C_{R,PV}$ = 1000W (standard panel capacity)
- $G_{T,STC}$ = 1000 W/m^2 (standard test conditions)
- $T_{C,STC}$ = 25°C (standard temperature)
- α_p = -0.4%/°C (temperature coefficient)

Web Interface Integration

The prediction system connects to a web-based interface through RESTful API endpoints:

- **Real-Time Endpoint:** Serves current weather data and immediate predictions
- **Forecast Endpoint:** Provides multi-day prediction arrays with confidence intervals
- **Model Status Endpoint:** Reports system health and last update timestamps

Scalability and Modularity Design

The architecture supports future enhancements through modular design:

- **Model Registry:** Enables easy integration of additional forecasting architectures
- **Geographic Scaling:** Supports multiple weather stations and regional models
- **Feature Expansion:** Accommodates new meteorological parameters or satellite data
- **Performance Monitoring:** Built-in logging and metric tracking for continuous improvement

This integrated architecture leverages the computational efficiency of GRU networks for immediate forecasting needs while utilizing Autoformer's advanced temporal modeling capabilities for strategic planning applications, creating a comprehensive solution for solar energy prediction across multiple operational time scales.

6.2.2 Short-Term Forecasting: GRU Architecture

Mathematical Foundation

The GRU network serves as our primary architecture for detecting short-term temporal patterns in weather data and generating next-day solar energy predictions.

The GRU operates through four sequential stages at each time step, utilizing the following key variables and operations:

Variables and Operations

Variables:

- x_t = current input at time step t (weather data)
- h_{t-1} = previous hidden state (memory from last time step)
- r_t = reset gate output (values between 0-1)
- u_t = update gate output (values between 0-1)
- h'_t = candidate hidden state (proposed new memory)
- h_t = current hidden state (updated memory)

Weight Matrices:

- W_r, W_u, W = weights for processing current input
- U_r, U_u, U = weights for processing previous hidden state
- b_r, b_u, b = bias terms

Operations:

- $\odot$ = element-wise multiplication (Hadamard product)

  e.g.:

  A = [1, 2, 3]

  B = [4, 5, 6]

  A $\odot$ B = [1×4, 2×5, 3×6] = [4, 10, 18]

- σ = sigmoid function: $\sigma(x) = \frac{1}{1+e^{-x}}$ (outputs 0-1)

- tanh = hyperbolic tangent: $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ (outputs -1 to 1)

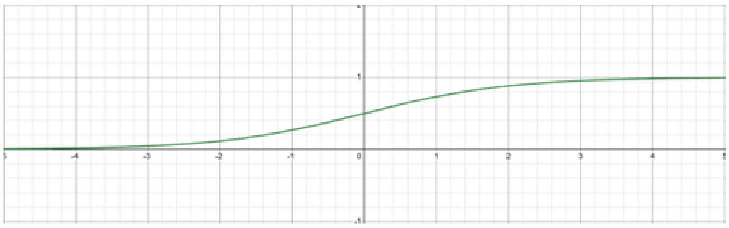

Figure 2: Sigmoid function graph

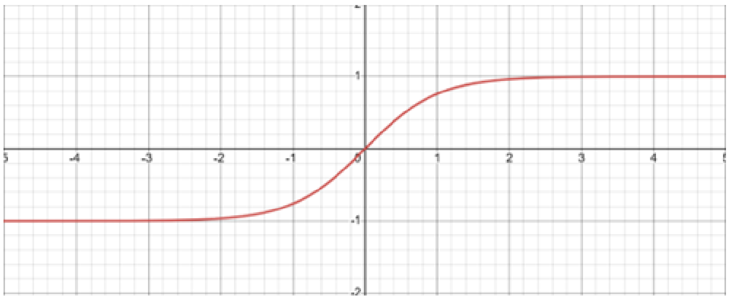

Figure 3: Hyperbolic tangent function graph

Processing Algorithm

GRU Computational Stages

The GRU processes sequential weather data through four computational stages:

1. **Reset Gate Calculation:** Determines how much previous information to forget

$$r_t = \sigma(W_r \cdot x_t + U_r \cdot h_{t-1} + b_r)$$

2. **Update Gate Calculation:** Controls balance between new and previous information

$$u_t = \sigma(W_u \cdot x_t + U_u \cdot h_{t-1} + b_u)$$

3. **Candidate Hidden State:** Generates new memory candidate using reset gate

$$h'_t = \tanh(W \cdot x_t + U \cdot (r_t \odot h_{t-1}) + b)$$

4. **Final Hidden State:** Combines previous and candidate states via update gate

$$h_t = u_t \odot h_{t-1} + (1 - u_t) \odot h'_t$$

This algorithm repeats for each time step in the weather data sequence, with the final hidden state processed through dense layers to generate solar energy predictions.

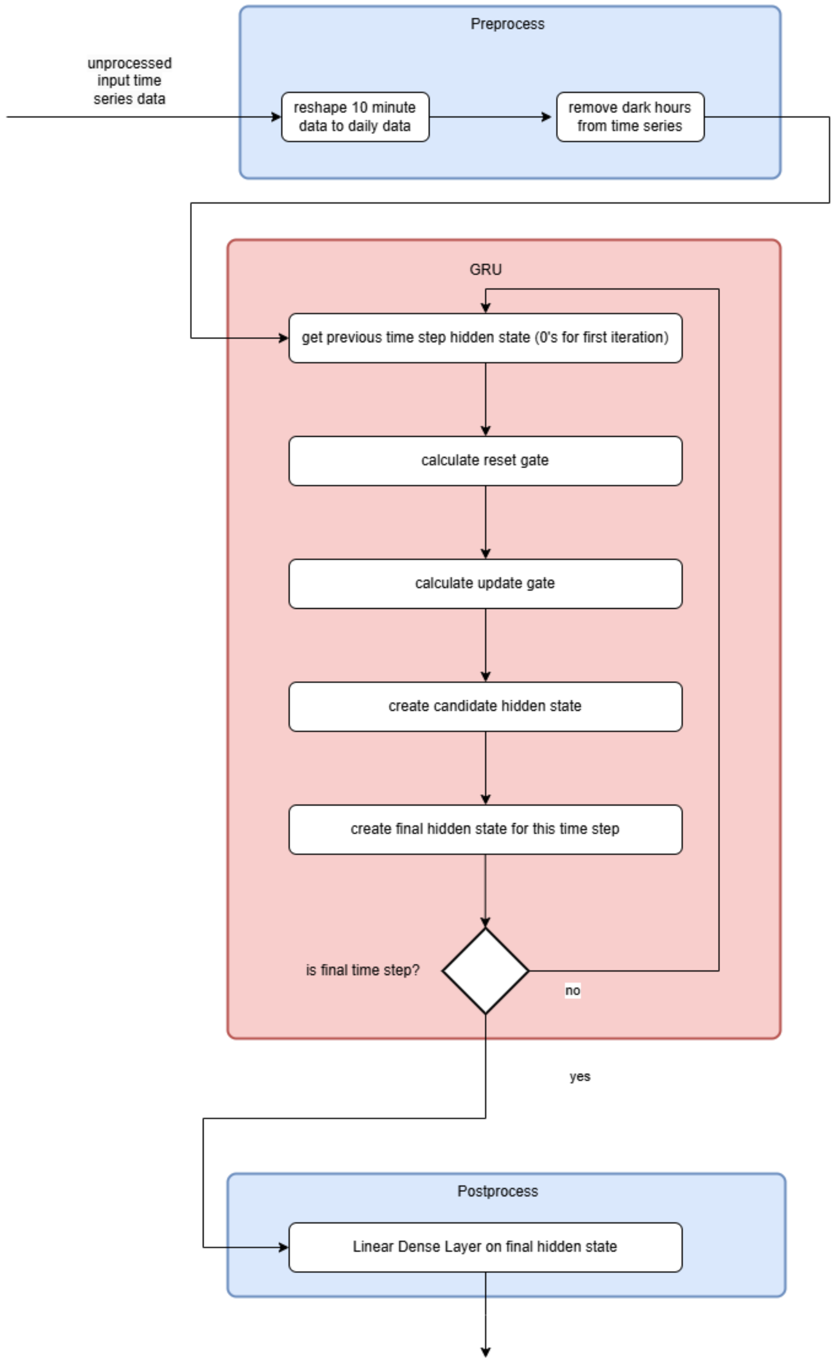

Figure 4: GRU Architecture Diagram

Implementation Specifications

GRU Network Architecture
The GRU model processes sequential weather data to generate next-day hourly predictions through the following architecture:

- **Input Layer:** Accepts normalized meteorological features (temperature, solar radiation, humidity, pressure)
- **GRU Layers:** Two stacked GRU layers with 64 and 32 hidden units respectively
- **Dropout Regularization:** 20% dropout between GRU layers to prevent overfitting
- **Dense Layers:** Fully connected layers for pattern integration and output generation
- **Output Layer:** Single neuron with linear activation for solar energy prediction

6.2.3 Long-Term Forecasting: Autoformer Architecture

Encoder-Decoder Architecture Design
The Autoformer architecture serves as our specialized transformer variant for extended solar energy prediction, utilizing autocorrelation mechanisms to capture seasonal and trend patterns in weather data.

The Autoformer follows a dual-module design optimized for long-term time series forecasting:

Encoder Module Processing
The encoder processes historical weather data through stacked identical layers to build comprehensive pattern understanding. Each encoder layer follows this processing flow:

1. **Input Processing:** Receives output from previous layer or initial embeddings
2. **Auto Correlation:** Applies correlation-based attention mechanism to the full input signal to identify dependencies and periodic patterns across the entire sequence
3. **Series Decomposition:** Splits the correlated output into trend components (gradual changes) and seasonal components (repeating patterns)
4. **Feed-Forward Processing:** Standard transformer layer applied to the seasonal component for additional pattern learning
5. **Series Decomposition:** Second decomposition step that further refines the separation of trend and seasonal components
6. **Component Handling:** Trend components are accumulated while seasonal components continue through the processing pipeline
7. **Output Generation:** Produces enhanced representation for subsequent layers

Decoder Module Processing

The decoder generates future predictions using encoder knowledge and progressive forecasting. Each decoder layer implements:

1. **Input Processing:** Receives output from previous layer or seasonal and trend initialization components
2. **Auto-Correlation:** First correlation operation on the seasonal component to capture self-dependencies
3. **Series-Decomposition:** Splits decoder sequence into trend and seasonal components
4. **Auto-Correlation:** Second correlation operation (cross-attention with encoder) to incorporate encoder knowledge
5. **Series-Decomposition:** Second decomposition to further separate trend and seasonal components
6. **Feed-Forward Processing:** Standard transformer layer applied to the seasonal component
7. **Series-Decomposition:** Final decomposition step
8. **Component Integration:** Throughout the process, trend components are accumulated and added together, while seasonal components flow through the processing chain
9. **Output Generation:** Produces predictions or enhanced representations

Key Processing Mechanisms

Auto-Correlation Processing (Seasonal): A specialized technique for identifying repeating patterns by measuring data similarity at different time delays. The mechanism:

- Tests correlation between data and time-shifted versions
- Uses Fast Fourier Transform for efficient correlation calculation
- Focuses attention on top-K strongest correlations
- Captures cyclical weather patterns like daily solar radiation cycles

Moving Average Pooling (Trend): A smoothing technique that reveals underlying trends by:

- Replacing data points with neighborhood averages
- Using sliding windows of configurable kernel sizes
- Filtering noise while preserving long-term patterns
- Reducing computational complexity through data compression

Cross-Attention Mechanism: Enables information exchange between encoder and decoder through:

- Query (Q): Decoder information representing needed correlations
- Key (K): Encoder pattern information of matching structure
- Value (V): Actual encoder data for informed predictions
- Attention scores calculated via Q·K operations with softmax weighting

The mathematical flow follows: For each decoder position, calculate attention scores $Q_1 \cdot K_1$, $Q_2 \cdot K_2$, $Q_3 \cdot K_3$, apply softmax normalization to obtain weights, then compute weighted sum of values for focused information retrieval.

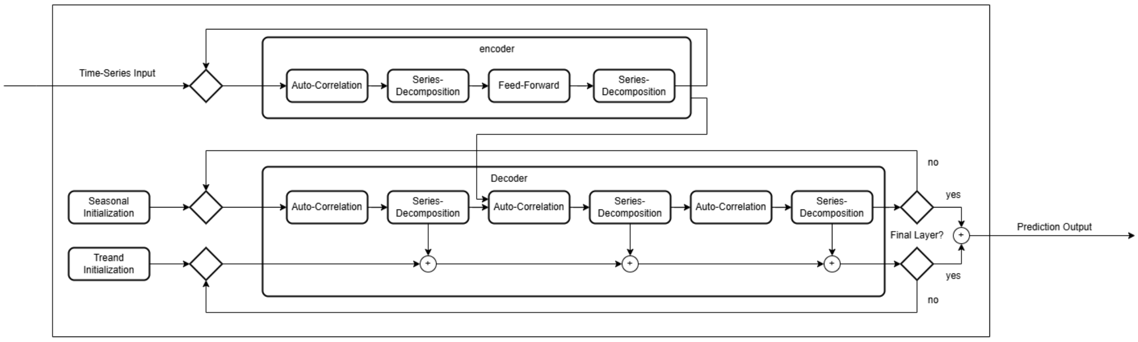

Figure 5: Autoformer Architecture Diagram

Implementation Specifications

Autoformer Network Architecture
The Autoformer architecture captures seasonal patterns and extended dependencies:

- **Encoder Stack:** 4 encoder layers with auto-correlation and decomposition mechanisms
- **Decoder Stack:** 2 decoder layers for progressive prediction generation
- **Embedding Dimension:** 128-dimensional representations for weather features
- **Attention Heads:** 8 attention heads for multi-scale pattern recognition
- **Decomposition Module:** Separates trend and seasonal components in weather data

6.2.4 Model Configuration and Evaluation

Hyperparameter Configuration

Our dual-architecture approach requires systematic optimization of key model parameters to achieve optimal forecasting performance. Based on established deep learning practices for time series forecasting, we have identified the most critical hyperparameters for tuning during implementation.

GRU Model Key Parameters

- **Hidden Units:** 32-128 neurons for temporal pattern learning
- **Sequence Length:** 24-72 hours of historical weather data
- **Learning Rate:** 0.001-0.01 with Adam optimizer
- **Dropout Rate:** 0.1-0.3 for overfitting prevention

Autoformer Model Key Parameters

- **Encoder/Decoder Layers:** 2-6 layers based on complexity needs
- **Attention Heads:** 4-16 heads for multi-scale pattern recognition
- **Embedding Dimension:** 64-256 for feature representation
- **Prediction Horizon:** 2-4 days ahead forecasting

Optimization Strategy
We will use grid search validation to identify optimal parameter combinations, with validation MAPE as the primary selection criterion. Parameters will be tuned systematically during model development to achieve our target 10% error margin for day-ahead predictions.

Model Evaluation and Validation Framework

Our forecast evaluation methodology employs a comprehensive multi-metric approach designed to validate the accuracy of our dual-architecture solar energy prediction system and demonstrate achievement of our target 10% error margin for day-ahead forecasts.

Primary Evaluation Metrics

We implement four primary evaluation metrics, each addressing different aspects of prediction quality:

Mean Absolute Error (MAE):

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

where y_i represents actual solar energy output and $\hat{y}_i$ represents predicted values. MAE provides interpretable error measurements in watts, enabling direct assessment of prediction accuracy in real-world units.

Root Mean Square Error (RMSE):

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

RMSE penalizes large prediction errors more heavily than small errors, making it particularly valuable for identifying model weaknesses during extreme weather conditions that significantly impact solar generation.

Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%$$

MAPE directly measures our target performance criterion, enabling verification of the 10% error margin goal. This scale-independent metric facilitates comparison across different solar installation capacities and seasonal conditions.

Coefficient of Determination (R^2):

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

where $\bar{y}$ represents the mean of actual values. R^2 quantifies the proportion of variance explained by our models, providing comparison benchmarks with existing research including Jebli et al. [1] who achieved R^2 = 92.61% for real-time and 93% for daily predictions.

Time Series Cross-Validation Strategy

Due to the temporal nature of weather data, we implement time series-specific validation techniques rather than random sampling:

Walk-Forward Validation:

- **Training Window:** 18 months of historical weather data
- **Validation Window:** 3 months rolling validation period
- **Test Window:** 3 months final evaluation period
- **Update Frequency:** Monthly model retraining with expanding window

Temporal Split Methodology:

- **Training Period:** January 2022 - June 2023
- **Validation Period:** July 2023 - September 2023
- **Test Period:** October 2023 - December 2023
- **Future Validation:** January 2024 onwards for live performance tracking

Established Forecasting Method Comparisons

We establish performance benchmarks through comparison with established forecasting methods:

Persistence Model: Assumes tomorrow's solar output equals today's output at corresponding times. This naive baseline provides minimum acceptable performance thresholds for operational forecasting systems.

Linear Regression Model: Multiple linear regression using meteorological parameters as independent variables and theoretical photovoltaic output as dependent variable, establishing traditional statistical modeling performance.

Moving Average Model: 7-day and 30-day moving averages of historical solar output, representing seasonal baseline approaches commonly used in energy planning.

Multi-Horizon Performance Assessment

Our dual-model approach requires evaluation across different forecasting horizons:

Short-Term Evaluation (GRU Model):

- **1-Hour Ahead:** Real-time operational accuracy for grid balancing
- **6-Hour Ahead:** Intraday trading and storage management decisions
- **24-Hour Ahead:** Day-ahead market participation and unit commitment

Long-Term Evaluation (Autoformer Model):

- **2-Day Ahead:** Weekend and maintenance planning accuracy
- **4-Day Ahead:** Weekly operational scheduling verification

Seasonal and Weather-Specific Analysis

Solar energy prediction accuracy varies significantly across different conditions, requiring specialized evaluation approaches:

Seasonal Performance Analysis:

- **Summer Performance:** High radiation, high temperature impact assessment
- **Winter Performance:** Low radiation, moderate temperature conditions
- **Transition Seasons:** Variable weather pattern handling capability

Weather Condition Stratification:

- **Clear Sky Days:** High predictability baseline performance
- **Partly Cloudy Days:** Variable radiation prediction accuracy
- **Overcast Days:** Low radiation forecasting capability
- **Extreme Weather:** Storm and unusual condition robustness

Theoretical Model Validation

Since direct solar farm data was unavailable, we validate our theoretical photovoltaic calculation approach:

Equation Validation: Our implementation of the photovoltaic output equation:

$$P_{out,pv} = C_{R,PV} \frac{G_T}{G_{T,STC}} [1 + \alpha_p (T_C - T_{C,STC})]$$

is validated against published solar energy research and industry standard calculations to ensure theoretical accuracy of our training labels.

Performance Targets and Acceptance Criteria

Our evaluation framework establishes clear success criteria:

- **Primary Target:** MAPE $\leq 10\%$ for day-ahead forecasts
- **Research Benchmark:** $R^2 \geq 90\%$ to match literature standards
- **Operational Target:** RMSE improvement $\geq 15\%$ over persistence model
- **Consistency Target:** Performance degradation $\leq 5\%$ across seasons

Future Monitoring Framework

While implementation of comprehensive monitoring systems extends beyond the scope of this academic project, our system architecture supports future deployment through modular design enabling:

- **Automated Performance Tracking:** System capability for daily accuracy calculation and trend analysis
- **Model Update Pipeline:** Framework design supports periodic retraining with expanding datasets
- **Seasonal Adaptation:** Architecture accommodates quarterly model recalibration for changing weather patterns
- **Scalability Considerations:** Modular design enables integration into larger operational forecasting systems

The proposed monitoring framework represents a natural extension for future research projects or industrial deployment, where continuous model improvement and adaptation become critical for maintaining long-term forecasting accuracy in operational solar energy systems.

This comprehensive evaluation framework ensures rigorous validation of our solar energy forecasting system while providing transparent documentation of model performance across multiple operational scenarios and time horizons within the constraints of an academic capstone project.

6.2.5 Algorithm Process

Master System Workflow

The main algorithm coordinates the entire forecasting process through a simple decision-making workflow:

1. **Data Collection:** Automatically fetch weather data from IMS API every 30 minutes
2. **Data Processing:** Clean, normalize, and prepare meteorological parameters
3. **Model Selection:** Choose appropriate model based on requested forecast horizon

4. **Prediction Generation:** Execute selected model to generate forecasts
5. **Output Delivery:** Format and serve predictions through web interface

Short-Term Prediction Pipeline

For next-day hourly predictions (0-24 hours ahead):

GRU Hourly Forecasting Algorithm

1. **Input Preparation:**
   - Collect last 48 hours of weather data
   - Normalize temperature, radiation, humidity, and pressure
   - Create time-based features (hour of day, season)
2. **Sequential Processing:**
   - Feed weather sequence through GRU layers
   - Process temporal patterns and dependencies
   - Generate hidden state representations
3. **Recursive Prediction:**
   - Predict first time step: $\hat{y}_{t+1}$
   - Use prediction as input for next time step: $\hat{y}_{t+2}$
   - Repeat for all time steps in time series
4. **Output Generation:**
   - Convert normalized predictions back to watts
   - Return hourly solar energy array for next day

Long-Term Prediction Pipeline

For multi-day predictions (1-4 days ahead):

Autoformer Daily Forecasting Algorithm

1. **Extended Input Preparation:**
   - Collect 7 days of historical weather data
   - Separate data into trend and seasonal components
   - Create positional encodings for temporal relationships
2. **Pattern Recognition:**

- Encoder processes historical patterns
- Identify recurring daily and weekly cycles
- Extract seasonal trends from weather data

3. **Multi-Day Prediction:**
   - Decoder generates predictions for days 1-4
   - Use autocorrelation to find similar past patterns
   - Predict daily average solar energy output
4. **Output Generation:**
   - Return daily solar energy predictions
   - Include confidence estimates based on pattern consistency

Model Selection and Routing Logic

The system automatically routes requests to the appropriate model:

Decision Process

1. **Analyze Request:** Determine requested forecast horizon
2. **Route to Model:**
   - If horizon $\leq$ 24 hours $\rightarrow$ Use GRU model
   - If horizon $>$ 24 hours $\rightarrow$ Use Autoformer model
3. **Execute Prediction:** Run selected model with current weather data
4. **Return Results:** Deliver predictions in requested format

Real-Time Processing and Optimization

For continuous operation and web interface updates:

Automated Processing Cycle

1. **Data Monitoring:**
   - Check IMS API for new weather data every 30 minutes
   - Validate data quality and completeness
2. **Update Trigger:**
   - Process new data when hourly updates detected
   - Update prediction cache with fresh forecasts

3. **Error Handling:**
   - Use previous predictions if API fails
   - Log errors for system monitoring
4. **Web Interface Update:**
   - Refresh dashboard with new predictions
   - Update accuracy tracking displays

Performance Optimization Strategies

To achieve our 30-second response time target:

- **Caching:** Store recent predictions for 30 minutes
- **Preprocessing:** Keep normalized data ready for immediate use
- **Model Loading:** Pre-load trained models in memory
- **Parallel Processing:** Handle multiple requests simultaneously

This streamlined algorithmic approach ensures efficient and accurate solar energy forecasting while maintaining clear separation between short-term and long-term prediction capabilities.

6.2.6 Future Development Roadmap

Our solar energy forecasting project follows a structured development timeline with clearly defined milestones and deliverables for each phase.

Phase 1: Data Processing

Data Pipeline Development

- **API Integration Enhancement:** Complete robust integration with Israel Meteorological Service (IMS) API, implementing error handling, data validation, and automatic retry mechanisms for reliable real-time weather data acquisition
- **Data Quality Assurance:** Develop comprehensive data cleaning procedures including outlier detection, missing value imputation using temporal interpolation techniques, and data consistency validation across multiple weather stations
- **Data Preprocessing Framework:** Implement Pearson correlation analysis for systematic meteorological parameter selection, create temporal feature encodings (hour-of-day, day-of-year, seasonal indicators), and establish data normalization pipelines for consistent model inputs

Solar Energy Label Generation

- **Photovoltaic Calculation Implementation:** Deploy the theoretical solar energy output calculation using the established equation: $P_{out,pv} = C_{R,PV} \frac{G_T}{G_{T,STC}}[1 + \alpha_p(T_C - T_{C,STC})]$ with comprehensive validation against published research standards
- **Dataset Creation:** Generate comprehensive training and validation datasets spanning multiple seasonal cycles with proper temporal stratification for time series modeling

Phase 2: Model Development and Training

GRU Model Implementation

- **Architecture Development:** Implement GRU network with optimized layer configuration (64-32 hidden units), dropout regularization (0.2), and appropriate sequence length (48-72 hours) for short-term solar energy forecasting
- **Training Pipeline:** Establish training procedures using Adam optimizer with learning rate scheduling, implement early stopping based on validation MAPE, and create model checkpointing for optimal weight preservation
- **Hyperparameter Optimization:** Conduct systematic grid search across key parameters including hidden units (32-128), dropout rates (0.1-0.3), and sequence lengths (24-72 hours) to achieve target 10% error margin

Autoformer Model Implementation

- **Architecture Deployment:** Implement Autoformer with 4 encoder layers, 2 decoder layers, 8 attention heads, and 128-dimensional embeddings optimized for long-term solar energy pattern recognition
- **Autocorrelation Mechanism:** Deploy specialized autocorrelation attention for seasonal pattern identification and moving average pooling for trend component extraction in weather data
- **Multi-Step Forecasting:** Establish direct multi-step prediction capability for 2-4 day ahead solar energy forecasting with comprehensive decomposition of trend and seasonal components

Phase 3: Model Integration and Evaluation

Dual-Model Integration

- **Pipeline Coordination:** Implement seamless integration between GRU and Autoformer models with appropriate temporal partitioning (GRU: 0-24 hours, Autoformer: 1-4 days)
- **Model Selection Logic:** Implement intelligent routing to determine optimal model based on requested forecast horizon
- **Performance Optimization:** Establish efficient inference pipelines for real-time prediction generation

Comprehensive Evaluation Framework

- **Metric Implementation:** Deploy comprehensive evaluation using MAE, RMSE, MAPE, and R^2 across multiple forecasting horizons with seasonal and weather-condition stratification
- **Baseline Comparisons:** Implement persistence, linear regression, and moving average models for performance benchmarking and validation of machine learning approach superiority
- **Cross-Validation:** Establish time series-specific walk-forward validation with 18-month training, 3-month validation, and 3-month test periods for robust performance assessment

Phase 4: Web Application Development

Backend Development

- **API Framework:** Develop RESTful API endpoints for real-time weather data, solar energy predictions, and model performance metrics using Flask or FastAPI framework
- **Database Integration:** Implement time series database (InfluxDB or PostgreSQL with TimescaleDB) for efficient storage and retrieval of weather and prediction data
- **Model Serving:** Deploy trained models with optimized inference pipelines, implement model versioning, and establish monitoring for prediction accuracy tracking

Frontend Development

- **Interactive Dashboard:** Create responsive web interface using React or Vue.js with real-time weather display, solar energy forecasts, and interactive prediction visualization charts

- **Performance Analytics:** Implement accuracy tracking displays, model comparison tools, and historical performance analysis with interactive visualizations
- **User Experience:** Design intuitive navigation, mobile-responsive layout, and accessible interface supporting both technical and non-technical users

Phase 5: Testing and Validation

System Integration Testing

- **End-to-End Validation:** Conduct comprehensive testing of data flow from IMS API through model inference to web interface display with real-time performance monitoring
- **Load Testing:** Validate system performance under concurrent user access and high-frequency prediction requests to ensure scalability and reliability
- **Accuracy Validation:** Perform extended validation using live weather data to demonstrate achievement of 10% MAPE target for day-ahead predictions

Documentation and Deployment

- **Technical Documentation:** Complete comprehensive system documentation including API specifications, model architecture details, and deployment procedures
- **User Documentation:** Create user guides, methodology explanations, and interpretation guidelines for forecast results and accuracy metrics
- **System Deployment:** Deploy production system with monitoring and logging procedures for reliable operational performance

Phase 6: Final Validation and Project Completion

Performance Validation

- **Live Testing:** Conduct extended live system operation with daily accuracy tracking and performance metric collection for final project evaluation
- **Comparative Analysis:** Generate comprehensive performance comparison with baseline methods and published research results to demonstrate project success
- **Seasonal Assessment:** Evaluate model performance across different weather conditions and seasonal variations to validate system robustness

Project Deliverables

- **Final Report:** Complete comprehensive project documentation including methodology, results, performance analysis, and recommendations for future development
- **System Demonstration:** Prepare live system demonstration showcasing real-time solar energy forecasting capabilities and accuracy achievements
- **Code Repository:** Deliver complete, documented codebase with installation instructions, configuration guides, and extension capabilities for future research

Success Criteria and Milestones

Each development phase includes specific success criteria and deliverable milestones:

Phase-Specific Success Metrics

- **Phase 1 Success:** Reliable data pipeline with $<2\%$ missing data and validated photovoltaic calculation accuracy
- **Phase 2 Success:** Trained models achieving $<15\%$ validation MAPE and successful architecture implementation
- **Phase 3 Success:** Integrated system achieving target 10% MAPE for day-ahead predictions and superior baseline performance
- **Phase 4 Success:** Functional web application with responsive interface and real-time prediction capabilities
- **Phase 5 Success:** Validated system performance with comprehensive testing and documentation completion
- **Phase 6 Success:** Final project delivery with demonstrated achievement of all primary objectives and targets

This structured roadmap ensures systematic development progression while maintaining flexibility to address challenges and optimize performance throughout the project lifecycle, ultimately delivering a robust solar energy forecasting system that meets our accuracy targets and provides valuable insights for renewable energy planning and operations.

6.2.7 Anticipated Challenges

The successful completion of this solar energy forecasting project faces several significant challenges stemming from both technical complexity and external circumstances. Understanding these challenges and developing mitigation strategies is crucial for project success within the constrained timeline.

Timeline and Resource Constraints

Challenge - Compressed Development Timeline: The project must be completed between August 2024 and January 2025, providing approximately 5 months for full implementation, testing, and validation. This compressed timeline coincides with the final year of our software engineering program and ongoing professional responsibilities.

Mitigation Strategies:

- **Agile Development Approach:** Implement iterative development with 2-week sprints to ensure continuous progress and early identification of bottlenecks
- **Minimum Viable Product (MVP):** Prioritize core functionality first, with advanced features implemented if time permits

Challenge - Military Reserve Duty Disruptions: As active reserve soldiers during wartime, both team members face potential military call-ups that could significantly impact project continuity. One team member has already been called to serve for two months during summer vacation (July 2024), demonstrating the unpredictable nature of military obligations.

Mitigation Strategies:

- **Comprehensive Documentation:** Maintain detailed technical documentation and code comments to enable seamless handover between team members
- **Modular Architecture Design:** Develop independent system components that can be worked on separately and integrated later
- **Version Control Strategy:** Implement rigorous Git workflow with frequent commits and detailed commit messages for project continuity
- **Backup Timeline:** Build 2-3 week buffer into project schedule to accommodate potential military service disruptions

Challenge - Work-Study Balance: Both team members maintain full-time employment while completing their final year of software engineering studies, creating significant time management challenges.

Mitigation Strategies:

- **Structured Time Allocation:** Dedicate specific evening hours (7-10 PM) and weekend sessions exclusively to project work
- **Employer Communication:** Inform employers about academic project requirements to potentially negotiate flexible working arrangements during critical development phases
- **Task Optimization:** Focus on high-impact activities during limited available hours, postponing non-critical features

Technical Implementation Challenges

Challenge - Data Availability and Quality: Access to real-time meteorological data from IMS API may face reliability issues, data gaps, or format inconsistencies that could impact model training and operational performance.

Mitigation Strategies:

- **Robust Data Pipeline:** Implement comprehensive error handling, data validation, and automatic retry mechanisms
- **Data Interpolation:** Develop temporal interpolation algorithms for handling missing weather measurements

Challenge - Model Complexity and Training Requirements: The dual-architecture approach using GRU and Autoformer models requires significant computational resources for training and optimization, potentially exceeding available hardware capabilities.

Mitigation Strategies:

- **Cloud Computing Utilization:** Leverage Google Colab Pro, AWS EC2, or Azure ML for high-performance model training when local resources are insufficient
- **Transfer Learning:** Utilize pre-trained time series models where possible to reduce training time and computational requirements

Challenge: Achieving Target Accuracy (10% MAPE) Meeting the ambitious accuracy target of 10% Mean Absolute Percentage Error for day-ahead predictions requires extensive hyperparameter optimization and model refinement.

Mitigation Strategies:

- **Systematic Hyperparameter Search:** Implement automated hyperparameter optimization using Optuna or similar frameworks
- **Baseline Validation:** Establish realistic accuracy expectations by implementing and comparing with simpler baseline models

Integration and Deployment Challenges

Challenge - Web Application Development: Creating a professional web interface while focusing primarily on machine learning model development may strain available development time and expertise.

Mitigation Strategies:

- **Framework Selection:** Use rapid development frameworks (Flask + Bootstrap, or React templates) to accelerate UI development
- **Progressive Enhancement:** Start with basic functionality and enhance interface iteratively
- **Template Utilization:** Adapt existing dashboard templates rather than building from scratch

Challenge - System Integration and Testing: Coordinating multiple system components (data pipeline, ML models, web interface) requires comprehensive integration testing under time constraints.

Mitigation Strategies:

- **Incremental Integration:** Test system components independently before full integration
- **Monitoring Implementation:** Build logging and monitoring into the system from the beginning for easier debugging

Risk Management and Contingency Planning

To address these challenges comprehensively, we have established a multi-tiered risk management approach:

High-Priority Contingencies:

- **Military Service Backup:** If either team member is called to extended military service, the remaining member will focus on completing core functionality with simplified scope
- **Technical Failure Backup:** If advanced models (Autoformer) prove too complex to implement effectively, focus on optimizing GRU model for all forecasting horizons

Success Probability Enhancement:

- **Regular Supervisor Consultation:** Bi-weekly meetings with project supervisors to ensure direction alignment and receive guidance on technical challenges
- **Industry Mentorship:** Leverage professional networks for guidance on machine learning implementation best practices

This comprehensive challenge assessment and mitigation strategy ensures project resilience against both predictable constraints and unexpected disruptions, while maintaining focus on delivering a functional solar energy forecasting system that meets our primary objectives within the available timeframe.

6.2.8 Tools and Components for the project

Development Environment and Version Control

- **Python:** Primary programming language for machine learning and backend development
- **Git & GitHub:** Version control and collaborative development platform
- **Visual Studio Code:** Integrated development environment with Python extensions
- **Jupyter Notebooks:** Interactive development for data analysis and model experimentation

Machine Learning and Data Processing

- **TensorFlow/Keras:** Deep learning framework for GRU and Autoformer implementation
- **Pandas & NumPy:** Data manipulation and numerical computing libraries
- **Scikit-learn:** Machine learning utilities for preprocessing and evaluation metrics
- **Matplotlib & Seaborn:** Data visualization and plotting libraries

Web Development

- **Flask:** Lightweight Python web framework for API development
- **HTML/CSS/JavaScript:** Frontend technologies for web interface
- **Bootstrap:** CSS framework for responsive web design
- **React.js:** JavaScript library for building interactive user interfaces and dynamic dashboard components
- **Chart.js:** JavaScript library for interactive data visualization

Data Sources and APIs

- **Israel Meteorological Service (IMS) API:** Primary weather data source
- **Requests Library:** HTTP library for API data retrieval
- **MongoDB Atlas (Free Tier):** Cloud-based NoSQL database for scalable weather data storage and historical records
- **PostgreSQL:** SQL database which could be enhanced with TimescaleDB for efficient time-series data manipulation

Cloud Services and Deployment

- **Google Colab (Free/Pro):** Cloud-based training environment for computationally intensive model training
- **Heroku (Free Tier):** Cloud platform for web application deployment
- **GitHub Pages:** Static site hosting for project documentation

Testing and Documentation

- **Pytest:** Python testing framework for unit and integration tests
- **LaTeX:** Document preparation system for project reports
- **Sphinx:** Documentation generator for code documentation

All listed tools are either completely free or offer sufficient free tiers for academic project requirements, ensuring cost-effective development while maintaining professional-grade capabilities.

7 System Requirements and Design

7.1 Requirements

Functional Requirements

ID	Requirement	Description
FR-1	Data Collection	The system shall collect weather data from external meteorological services.
FR-2	Data Preprocessing	The system shall clean and preprocess collected data for model input.
FR-3	Solar Energy Calculation	The system shall calculate theoretical solar energy output using photovoltaic equations.
FR-4	Short-Term Forecasting	The system shall generate next-day solar energy predictions.
FR-5	Long-Term Forecasting	The system shall provide multi-day solar energy predictions.
FR-6	Parameter Selection	The system shall identify relevant meteorological parameters for prediction.
FR-7	Model Training	The system shall train machine learning models on historical data.
FR-8	Performance Evaluation	The system shall calculate accuracy metrics and evaluate model performance.
FR-9	Data Visualization	The system shall display forecasts and performance metrics through interactive charts.
FR-10	Web Interface	The system shall provide a web-based user interface for accessing forecasts.
FR-11	Real-Time Updates	The system shall update forecasts automatically when new data is available.
FR-12	Historical Analysis	The system shall track prediction accuracy over time.
FR-13	Documentation Access	The system shall provide access to methodology and technical documentation.

Non-Functional Requirements

ID	Requirement	Description
NFR-1	Prediction Accuracy	The system shall achieve forecast accuracy with error margin not exceeding 10% for day-ahead predictions.
NFR-2	Response Time	The system shall generate forecasts within maximum response time of 30 seconds per request.
NFR-3	Scalability	The system shall handle data from multiple weather stations without performance degradation.
NFR-4	Maintainability	The system shall be designed with modular architecture to facilitate maintenance and updates.
NFR-5	Usability	The web interface shall be intuitive and accessible to non-technical users.
NFR-6	Resource Efficiency	The system shall optimize computational resource usage for training and inference operations.
NFR-7	Extensibility	The system shall allow integration of additional models with minimal architectural changes.
NFR-8	Security	The system shall implement secure data handling and API access management.
NFR-9	Compatibility	The system shall be compatible with major web browsers and mobile devices.
NFR-10	Data Processing	The system shall process weather data updates within 30 minutes of availability.
NFR-11	Storage	The system shall store at least 2 years of analytical data.

7.2 Frontend Interface Design

Frontend Page Specifications

Dashboard Page (Main Interface)

The primary landing page displaying current solar energy predictions and quick navigation to detailed forecasting sections. Features include selecting a time frame and the number of panels.

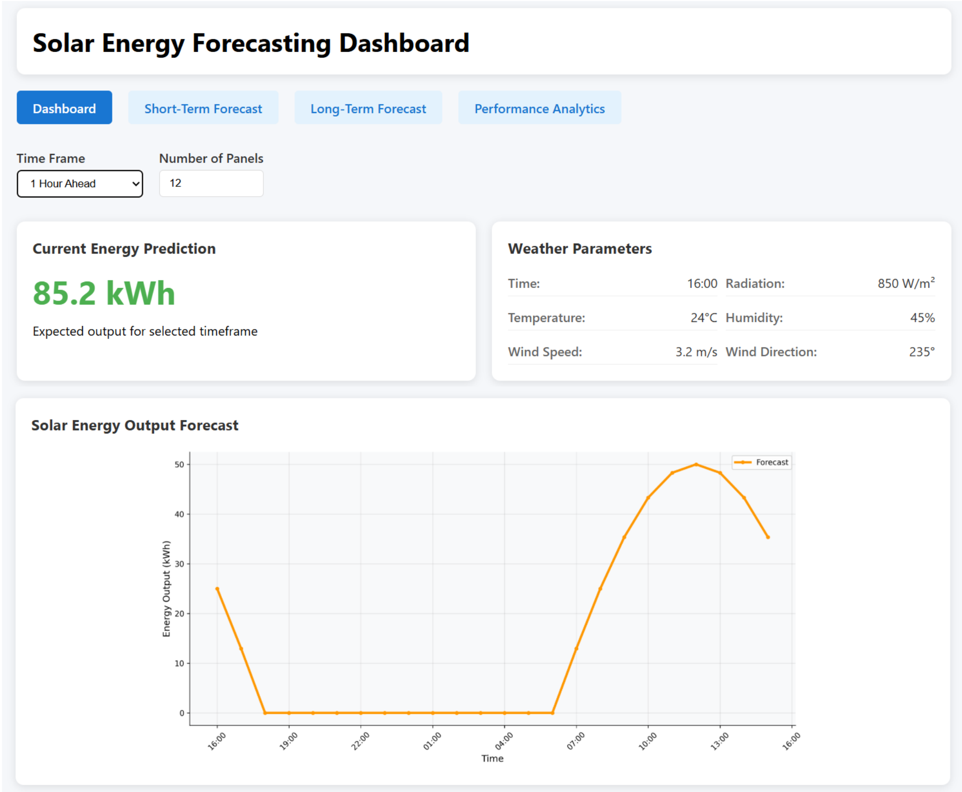

Figure 6: Dashboard Page Interface

Short-Term Forecasting Page

Dedicated interface for detailed next-day predictions using the GRU model. Displays hourly solar energy output predictions for the next 24 hours through interactive line charts, confidence intervals for each hourly prediction, and comparison toggle with previous day's actual vs predicted values.

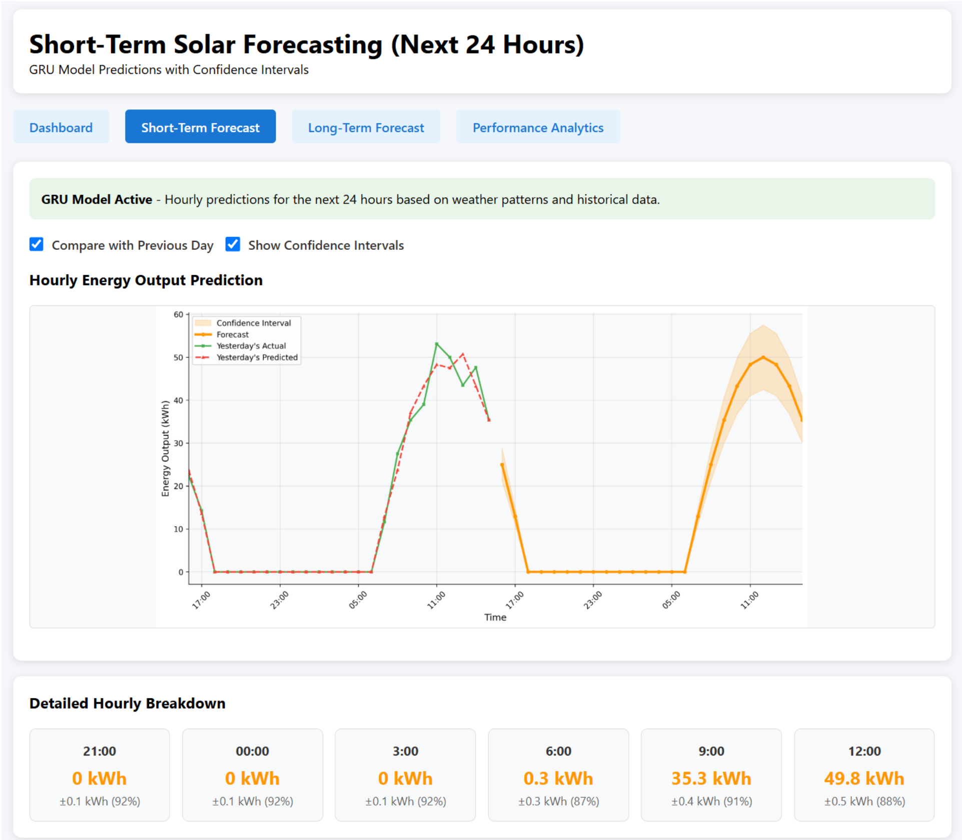

Figure 7: Short-Term Forecasting Page Interface

Long-Term Forecasting Page

Extended prediction interface utilizing the Autoformer model for multi-day forecasting. Shows 2-4 day ahead daily solar energy predictions, seasonal trend analysis charts, and weather pattern correlation displays.

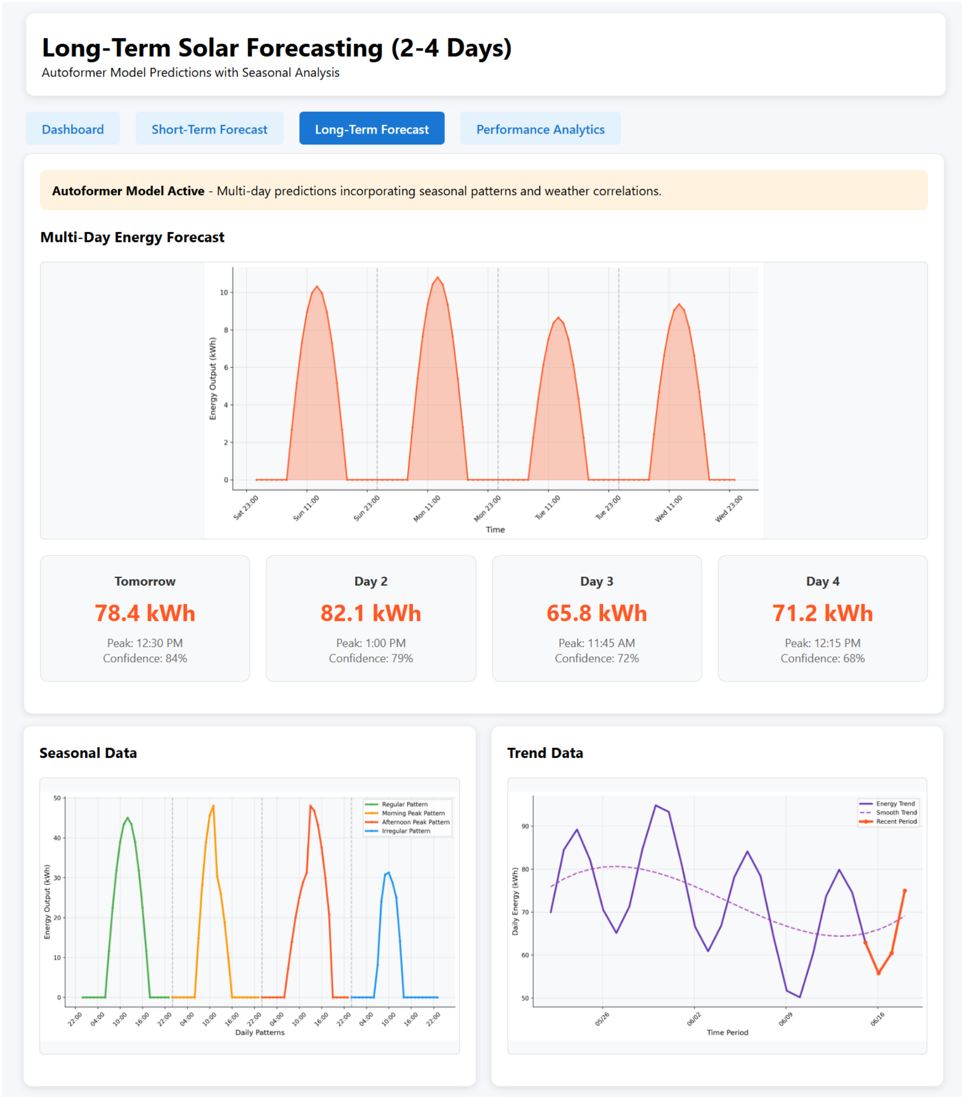

Figure 8: Long-Term Forecasting Page Interface

Performance Analytics Page

Model accuracy tracking and validation interface for technical users. Contains historical prediction accuracy metrics (MAE, RMSE, MAPE), model performance comparison charts between GRU and Autoformer, and seasonal accuracy variation analysis.

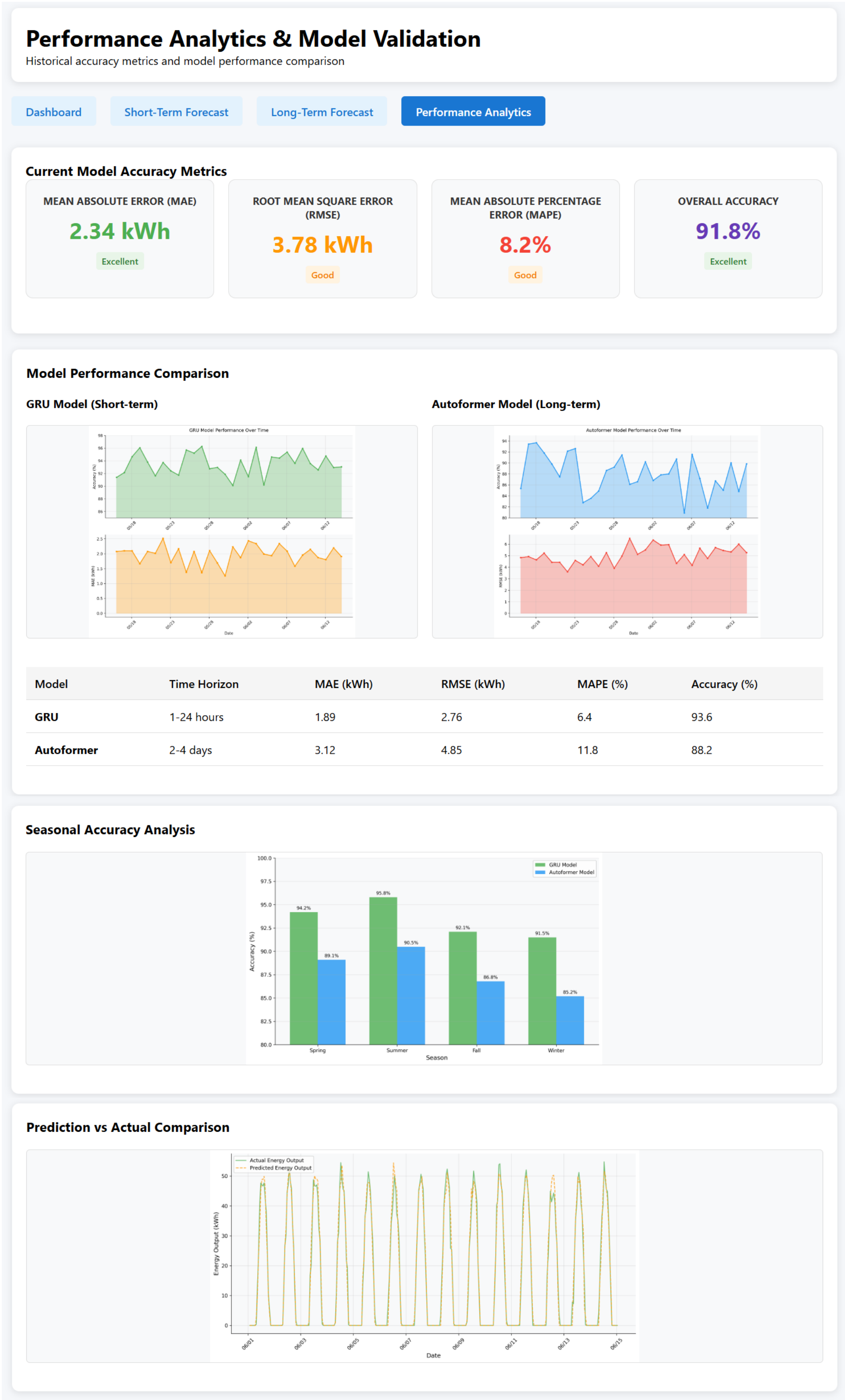

Figure 9: Performance Analytics Page Interface

7.3 Activity Diagram

The activity diagram illustrates the key user workflows and system processes for solar energy forecasting, including data collection, model selection logic, and prediction generation flows.

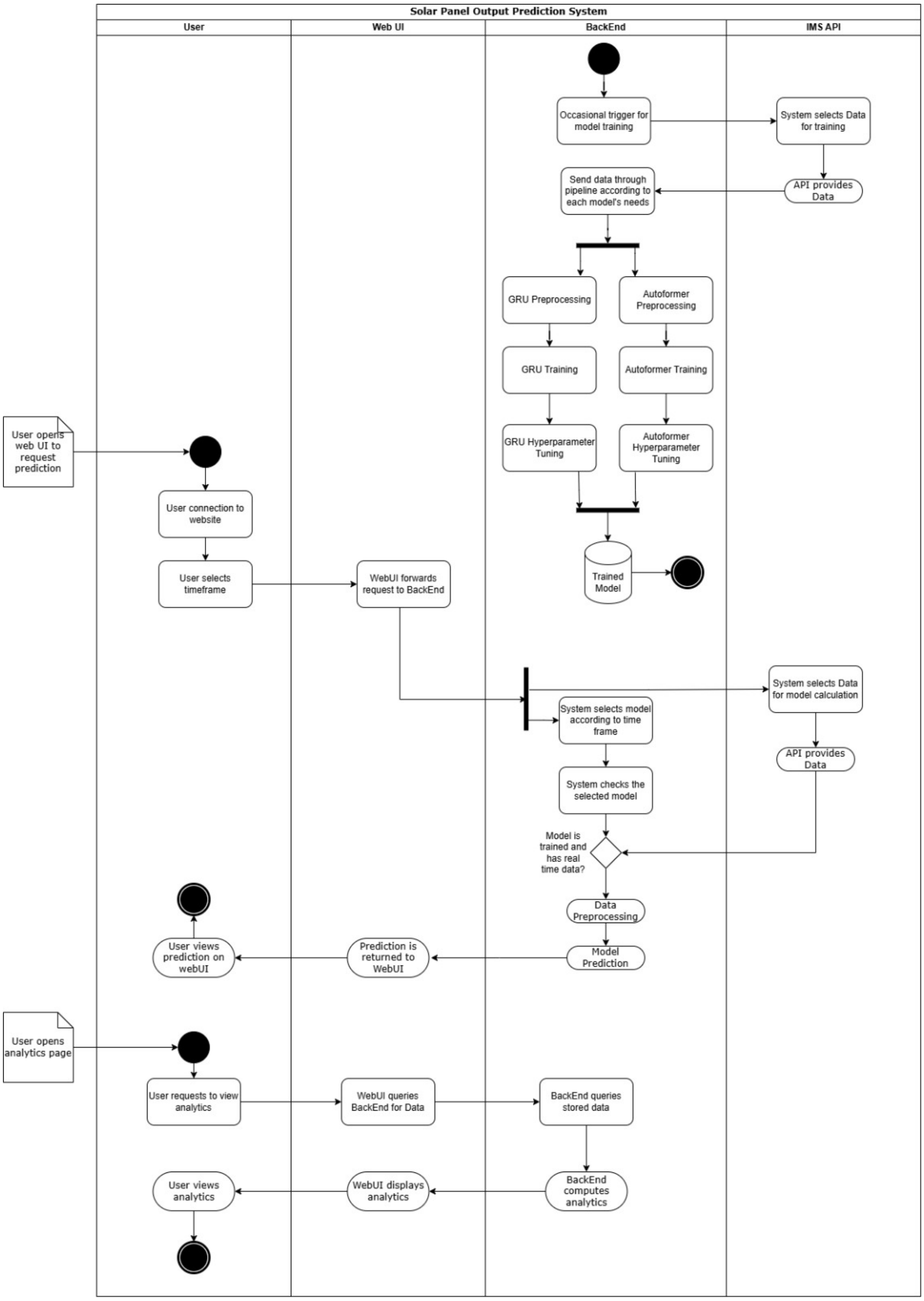

Figure 10: Key Feature Flows - Solar Energy Forecasting System

The diagram focuses on primary user interactions: accessing real-time forecasts, generating short-term predictions via GRU model, and creating long-term forecasts using the Autoformer architecture.

8 References

References

[1] Jebli, I., Belouadha, F. Z., Kabbaj, M. I., & Tilioua, A. (2021). Prediction of solar energy guided by Pearson correlation using machine learning. Energy, 224, 120109.

[2] Casolaro, A., Capone, V., Iannuzzo, G., & Camastra, F. (2023). Deep Learning for Time Series Forecasting: Advances and Open Problems. Information, 14(11), 598.

[3] Hugging Face. (2023). Yes, Transformers are Effective for Time Series Forecasting (+ Autoformer). Retrieved from https://huggingface.co/blog/autoformer

[4] Scientific Reports. (2024). A novel extreme adaptive GRU for multivariate time series forecasting. Nature Scientific Reports. Retrieved from https://www.nature.com/articles/s41598-024-53460-y

[5] MDPI. (2023). Short-Term Photovoltaic Power Forecasting Based on a Novel Autoformer Model. Symmetry, 15(1), 238.

[6] Wu, H., Xu, J., Wang, J., & Long, M. (2021). Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. OpenReview. Retrieved from https://openreview.net/forum?id=J4gRj6d5Qm

[7] Scientific Reports. (2025). Novel model for medium to long term photovoltaic power prediction using interactive feature trend transformer. Nature Scientific Reports. Retrieved from https://www.nature.com/articles/s41598-025-90654-4

[8] Artificial Intelligence Review. (2024). A systematic review for transformer-based long-term series forecasting. Springer. Retrieved from https://link.springer.com/article/10.1007/s10462-024-11044-2