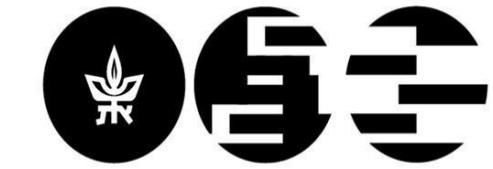


Improving Solar Power Forecasting Using Neural Networks

Project Number: **3022**

Project Report

Student: Hadar Levy ID: 322280645

Student: Gal Schwartz ID: 208308874

Supervisor: Khen Cohen

Project Carried Out at: Tel Aviv University

Table of Contents

List of Figures

List of Tables

List of Equations

Abstract

This project addresses the challenge of short-term solar irradiance forecasting, focusing specifically on Global Horizontal Irradiance (GHI), which is a key metric for predicting energy production from photovoltaic (PV) cells. Accurate short-term GHI predictions are crucial for optimizing power grid operations, reducing reliance on backup systems, and improving the integration of solar energy into national energy infrastructures.

The purpose of the project is to develop a robust, data-driven forecasting model that accounts for both temporal and spatial dependencies in solar irradiance data. To that end, we propose a hybrid deep learning architecture that integrates Graph Attention Networks (GAT) with Long Short-Term Memory (LSTM) networks. This model processes recent irradiance time series from multiple PV sites and captures localized weather dynamics and cloud movement across the spatial grid.

The project deliverable is a trained and validated spatiotemporal model capable of forecasting GHI 15, 30, 45, and 60 minutes ahead. The system accepts real-time GHI measurements from a network of PV sites, represents them as a graph, and produces accurate irradiance forecasts for short-term operational use. The model's performance was evaluated using two standard metrics in solar forecasting: Normalized Root Mean Square Error (NRMSE) and Normalized Mean Absolute Error (NMAE).

The following Figure presents the end-to-end block diagram and potential integration into real-time solar forecasting systems.

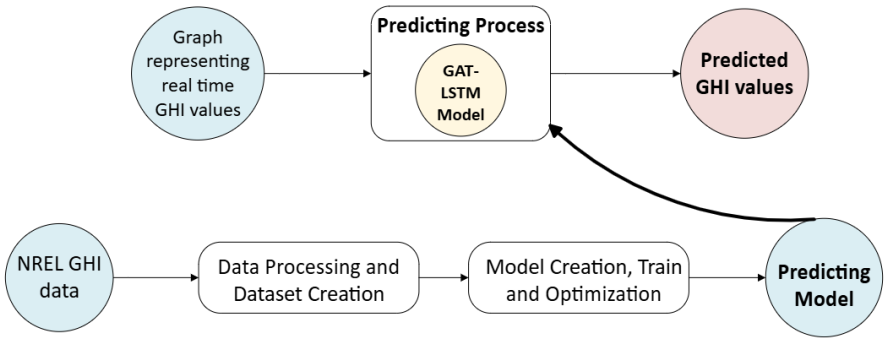


Figure 1. End-to-end block diagram including the project's working flow and integration into real-world application.

1 Introduction

1.1 Motivation

Over the past decade, solar energy has become a cornerstone of global efforts to transition toward sustainable and renewable sources of electricity. Among the various renewable technologies, PV systems have experienced particularly rapid growth due to decreasing costs, ease of deployment, and increasing efficiency. However, one of the most significant challenges facing the widespread integration of solar power into modern energy systems is its inherent intermittency and dependence on weather conditions.

Solar irradiance, and particularly GHI, varies significantly due to dynamic atmospheric phenomena such as cloud movement, aerosol concentration, humidity, and other meteorological variables [1]. These fluctuations can lead to severe deviations between expected and actual solar energy production, making it difficult to ensure grid reliability and economic efficiency. Inaccurate forecasts may result in inefficient energy dispatch, over-reliance on backup systems, and increased operational costs.

To address this issue, there is a growing need for robust and accurate GHI forecasting models capable of predicting short-term and intra-day solar irradiance levels. Traditional statistical and physical models, while useful in certain contexts, often struggle to capture the complex spatiotemporal relationships inherent in solar radiation data. Recent advances in machine learning, and particularly deep learning, have shown promise in improving forecast accuracy by learning complex nonlinear dependencies from large datasets [2].

1.2 Project Goals

The primary objective of this project is to forecast GHI at 15, 30, 45, and 60 minutes using a deep learning model. To achieve this, we develop spatiotemporal architecture that predicts GHI across a network of PV sites by learning both temporal patterns within each site and spatial interactions between sites. The proposed approach aims to improve the accuracy of short-term GHI forecasts, especially under rapidly changing weather conditions.

1.3 Approach

To address the challenge of accurate short-term solar irradiance forecasting, we propose a deep learning architecture that integrates GAT with LSTM units. This hybrid model is specifically designed to capture both spatial dependencies across multiple PV sites and temporal patterns within each site's irradiance history.

In our approach, the network of PV systems is modeled as a graph, where each node represents a PV site and edges encode spatial relationships, such as geographical proximity. At each node, a sequence of recent GHI measurements is processed using LSTM layers to learn temporal dynamics. The encoded temporal features are then passed through a GAT layer, which adaptively aggregates information from neighboring sites, allowing the model to focus on relevant spatial contexts - such as nearby sites affected by the same cloud patterns. This spatiotemporal modeling approach enables the network to account for cloud movement, localized weather changes, and site-specific temporal trends.

The final model produces GHI forecasts for multiple future time steps, offering improved accuracy and robustness, especially in dense PV site networks where spatial information plays a crucial role.

1.4 Comparison Against Existing Work

In recent years, a wide range of methods have been proposed for tackling the problem of solar irradiance forecasting. Many state of the art models integrate machine learning techniques - especially hybrid architectures that combine GNNs with other components such as CNNs, LSTMs, or attention-based mechanisms [2][3][4]. These models are typically designed and optimized for specific datasets, often collected in regions with different climate profiles.

Our work focuses on forecasting solar irradiance in Israel, a region with unique climatic conditions that are not adequately represented in most of the publicly available datasets used in previous research. For example, a related study was conducted in Switzerland using local meteorological and solar data [5]. However, the Switzerland climate is significantly different from that of Israel - characterized by more frequent rainfall and less consistent solar exposure. These climatic differences affect both the data distribution and the modeling requirements, which is why models developed for Switzerland cannot be directly applied to the Israeli context without adaptation.

Moreover, while many models in the literature aim to predict power output of PVs, our model directly forecasts GHI. In our project, we chose to forecast GHI rather than PV power output. This decision was based on two primary considerations. First, the dataset available to us consisted of simulated GHI measurements at predefined geographic points, rather than real-world PV system outputs. These locations do not necessarily correspond to actual PV installations. Second, predicting PV output requires additional information beyond irradiance - specifically, the physical characteristics of the solar panel systems, including panel size, orientation, tilt, efficiency, and system losses. In the absence of such data, forecasting GHI offers a more direct and generalizable approach, which can later be used as input for downstream PV yield estimation models where system parameters are known.

Additionally, there are some differences in architecture compared to other models. Although some high-level architectural similarities exist between our model and those found in literature (e.g. the use of GNNs to capture spatial dependencies), our implementation differs in several key aspects:

- **Data modeling**: We adapted the input representation and preprocessing steps to match the characteristics of local meteorological and solar data.
- **Network design**: We modified the structure of the GNN and its integration with temporal components to better capture local spatial-temporal dynamics.

Throughout the development process, we reviewed multiple academic works and implementations [3][4][5]. Rather than replicating an existing model, we selectively integrated techniques and design principles from various sources. These were adapted and modified to suit our goals, ensuring that the resulting model is both informed by current research and tailored to the unique properties of our dataset.

2 Theoretical Background

This section presents the theoretical foundations and algorithms relevant to the forecasting of GHI using deep learning methods. It includes a review of the core concepts behind spatiotemporal modeling, as well as the main components of the proposed architecture - GNNs (in particular GAT) and LSTM networks. In addition, useful mathematical and statistical approaches for GHI forecasting are discussed in this section.

2.1 Global Horizontal Irradiance (GHI)

GHI is the total amount of solar radiation received per unit area on a horizontal surface on earth. It is a key metric in solar energy forecasting and PV system performance analysis. GHI consists of two components:

- Direct Normal Irradiance (DNI) projected onto the horizontal plane, which comes directly from the sun without being scattered.
- Diffuse Horizontal Irradiance (DHI), which is solar radiation scattered by the atmosphere and clouds.

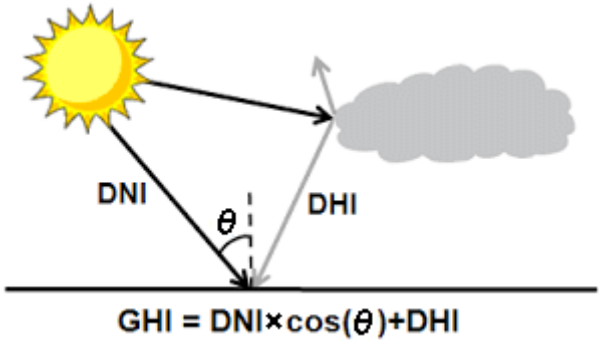


Figure 2. Components of GHI.

Mathematically, GHI can be expressed as described in Equation 1:

$$\text{Equation 1: } \boldsymbol{GHI = DHI + DNI \cdot cos(\theta z)}$$

Where:

- θz is the solar zenith angle (the angle between the sun and the vertical direction).
- DHI and DNI are measured in watts per square meter $[\frac{\mathrm{W}}{\mathrm{m}^2}]$.

GHI varies based on geographic location, time of day, season, and atmospheric conditions such as cloud cover and aerosols. Understanding GHI is critical for predicting PV power output, as it directly influences the amount of electricity generated by PV panels. The relation between GHI and PV power is linearly [6].

2.2 Long Short-Term Memory (LSTM) Networks

LSTM networks are a specialized form of Recurrent Neural Networks (RNNs) designed to effectively capture long-range dependencies in sequential data. While traditional RNNs are capable of processing time series inputs, they suffer from limitations such as the vanishing or exploding gradient problem, which makes it difficult for the network to learn long-term dependencies. LSTM networks address this issue using a unique internal structure composed of memory cells and gating mechanisms.

Each LSTM unit contains three primary gates:

- Input gate - which controls how much new information is allowed into the memory cell.
- Forget gate - which determines what information from the past should be discarded.
- Output gate - which regulates how much of the memory cell's content is exposed to the next layer or time step.

These gates allow LSTMs to retain relevant information for extended periods while discarding less useful data, making them well-suited for time series tasks where dependencies may span over hours, days, or even seasons.

In the context of GHI forecasting, LSTMs are used to analyze historical GHI measurements from a given PV site to extract recurring temporal patterns. These may include:

- Short-term weather fluctuations (e.g. transient clouds).
- Periodic behaviors influenced by the time of year (e.g. changes in sun angle across seasons).

By learning these temporal dependencies, LSTM can help generate more accurate short-term forecasts of solar irradiance. However, one of their limitations is that they process each site independently and do not capture the spatial interactions between neighboring PV locations, which are often critical due to moving cloud systems. This limitation motivates the integration

of LSTM with graph-based models such as GATs, which explicitly model spatial dependencies - leading to a more complete spatiotemporal forecasting framework.

2.3 Graph Neural Networks (GNNs)

2.3.1 Update Mechanism in a GNN Layer

GNNs are a class of deep learning models designed to process data that is represented in the form of graphs. Unlike conventional neural networks that operate on regular grid-like data structures (such as images or sequences), GNNs are capable of learning from irregular and relational structures, where the relationships between data points carry important meaning.

In a GNN, each node represents a data point (e.g. a PV site), and edges represent connections or relationships (e.g. geographic proximity). The key idea behind GNNs is that each node should update its internal representation (feature vector) not only based on its own features, but also by incorporating information from its neighbors in the graph. This is done through a process called message passing.

At each layer of the network, nodes perform two main steps:

1. **Aggregation -** Each node gathers feature information from its neighbors.
2. **Update -** The node combines the aggregated information with its current features to produce a new representation.

This process allows each node to iteratively build a richer understanding of its local and extended environment.

The node update at layer $k+1$ can be mathematically expressed as:

Equation 2: $$\boldsymbol{h_u}^{(\boldsymbol{k+1})} = \boldsymbol{UPDATE}^{(\boldsymbol{k})}(\boldsymbol{h_u}^{(\boldsymbol{k})}, \boldsymbol{AGGREGATE}^{(\boldsymbol{k})}(\{\boldsymbol{h_v}^{(\boldsymbol{k})} \mid \boldsymbol{v} \in \boldsymbol{N}(\boldsymbol{u})\}))$$

Where:

- $N(v)$ is the set of neighbors of node v.
- ${h_u}^{(k)}$ is the feature vector of neighbor u at layer k.
- *AGGREGATE* is a function such as sum, mean, or attention-weighted sum that combines the features of the neighboring nodes.
- *UPDATE* is typically a learnable function (e.g., a linear transformation with an activation function) that updates the node's features based on the aggregated information.

In GNN, each layer contains a learnable weight matrix $W^{(l)}$ that is shared across all nodes and edges within that layer. This matrix transforms the input features of neighboring nodes during message passing, allowing the model to learn how to extract meaningful information from a node's local neighborhood. However, the weight matrix is not shared across layers. Each layer has its own unique set of weights, enabling the network to progressively learn more abstract and complex representations as depth increases. This layer-wise independence of weights is essential for hierarchical learning in deep neural architectures. Figure 3 illustrates the GAT updating mechanism:

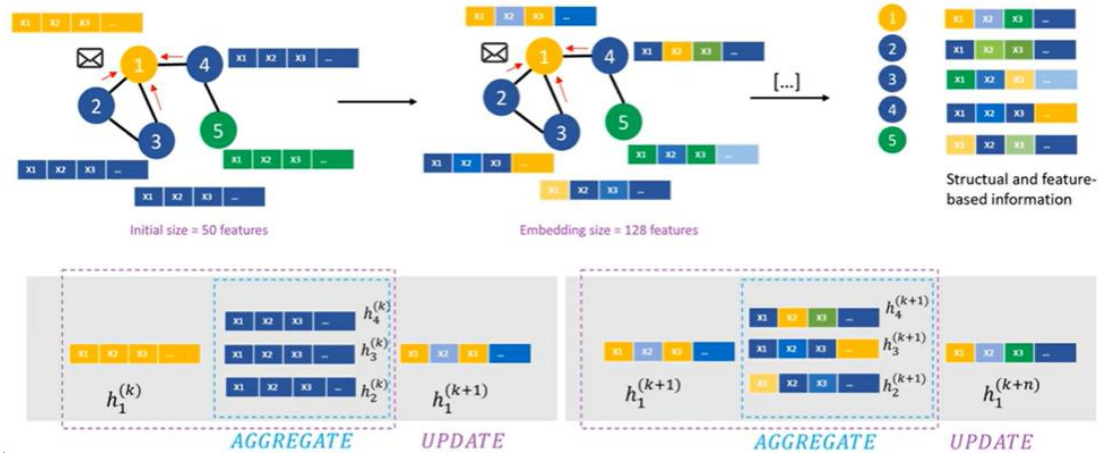


Figure 3. GAT updating mechanism.

An example for a specific message passing: for a node v, its updated representation $h_v{}^{(l+1)}$ at layer $l+1$ is computed as described in Equation 3:

Equation 3: $$\boldsymbol{h_v^{(l+1)} = \sigma\left(\sum_{u \in N(v)} W^{(l)} h_u^{(l)} + b^{(l)}\right)}$$

Where:

- $N(v)$ is the set of neighbors of node v.
- $h_u{}^{(l)}$ is the feature vector of neighbor u at layer l.
- $W^{(l)}$ is the weight matrix for layer l. It is shared across all neighbors and nodes in that layer. The purpose of this matrix is to learn how to transform the input features into a more useful representation for the task.
- $\sum$ is the aggregation step, where the transformed feature vectors of all neighbors are summed (or otherwise combined). It allows node v to collect information from its surrounding nodes.
- $b^{(l)}$ is the bias vector, which is added after aggregation, as in standard neural network layers.

- σ denotes an activation function (such as ReLU), which introduces non-linearity into the model and helps it learn complex patterns.

2.3.2 Depth Meaning in GNNs

The depth of a GNN refers to the number of message-passing layers in the network. Each additional layer allows nodes to access information from farther away in the graph.

- With 1 layer, a node gathers information from its immediate (1-hop) neighbors.
- With 2 layers, it can access its neighbors' neighbors (2-hop neighborhood).
- With k layers, each node effectively incorporates information from nodes up to k steps away in the graph.

2.4 Graph Attention Networks (GAT)

GATs are a type of GNN that introduces an attention mechanism that allows each node to assign different weights to different neighbors when aggregating their information. This makes GATs more flexible and context-aware, especially in situations where not all neighbors are equally important - which is common in real-world graphs, including GHI forecasting networks.

2.4.1 Update Mechanism in a GAT layer

The update of node representations in a GAT layer is defined by the following Equation:

Equation 4: $$\boldsymbol{h_v}^{\boldsymbol{l+1}} = \boldsymbol{\sigma}(\sum_{\boldsymbol{u}\in \boldsymbol{N}(\boldsymbol{v})} \boldsymbol{\alpha_{uv}}^{(\boldsymbol{l})} \boldsymbol{W h_u}^{(\boldsymbol{l})})$$

Where α_{uv} is the attention weight. By default, there is no bias term to prevent overfitting but can be added if desired.

2.4.2 Attention Score Computation

This score $e_{vu}^{(l)}$ represents how important neighbor u information is to node v. It's based on the content of their feature vectors - the network learns which neighbors are most useful based on their values, not just graph structure. The attention score between a target node v and its neighbor u in layer l is computed as follows:

Equation 5: $$\boldsymbol{e_{vu}^{(l)}} = \boldsymbol{LeakyReLU}(\boldsymbol{a^T}[\boldsymbol{W h_v}^{(\boldsymbol{l})} || \boldsymbol{W h_u}^{(\boldsymbol{l})}])$$

Where:

- $[Wh_v{}^{(l)} || Wh_u{}^{(l)}]$ is concatenation of the transformed feature vectors of nodes v and u.
- a is a learnable vector shared across all node pairs, which assigns a score to the concatenated representation. It learns how node u is important to node v.

Note that GAT does not perform the dot product between the two feature vectors because their similarity does not necessarily have correlation to the prediction.

2.4.3 Attention Coefficient Normalization

Equation 6: $\boldsymbol{\alpha_{vu}}^{(l)} = \frac{\boldsymbol{e_{vu}^{(l)}}}{\sum_{\boldsymbol{k \in N(v)}} \boldsymbol{e^{e^{(l)}}_{vk}}}$

Normalizing attention scores in GAT with SoftMax makes the aggregation process stable, interpretable, and independent of the number of neighbors. It ensures that each node combines information from its neighbors in a meaningful and well-scaled way.

For a detailed explanation of the attention mechanism in Graph Attention Networks we refer the reader to the original GAT paper by Veličković et al. (2018) [7].

2.5 Autocorrelation

Auto correlation is mathematical computation representing the similarity between a signal and a delayed copy of itself. The calculation is by the following Equation 7:

Equation 7: $\boldsymbol{\rho_k(t)} = \frac{\sum_{\boldsymbol{t=1}}^{\boldsymbol{N-k}} (\boldsymbol{x_{t+k}} - \overline{\boldsymbol{x}})(\boldsymbol{x_t} - \overline{\boldsymbol{x}})}{\sum_{\boldsymbol{t=1}}^{\boldsymbol{N}} (\boldsymbol{x_t} - \overline{\boldsymbol{x}})^2}$

3 Implementation

3.1 Flow and Block Diagram

The block diagram presented in the abstract (Figure 1) illustrates our end-to-end workflow for developing the forecasting model, as well as its integration into a potential real-world application. The system is designed to store historical GHI values for each PV site and use this data to construct a graph representation at each time step. This graph is then provided as input to a pre-trained GAT-LSTM model, which outputs future GHI predictions. This setup enables continuous, real-time forecasting by leveraging recent measurements and the learned spatial-temporal patterns captured during the training phase.

3.2 Pre-Processing

In this section we will discuss the pre-processing actions applied on the data in order to make it suitable to solve the problem efficiently.

3.2.1 Splitting

We utilized GHI data from the National Renewable Energy Laboratory (NREL). To ensure proper model training and evaluation, the dataset was split into two distinct subsets: training and testing.

- The **training set** consisted of GHI data from January to June 2018.
- The **test set** comprised data from the year 2017 from the 5/1/2017 to 18/1/2017.

This chronological separation ensures a realistic forecasting scenario by preventing data leakage - where future information could inadvertently influence model training. By maintaining strict separation between datasets, we improved the model's ability to generalize and accurately assess its performance on unseen data.

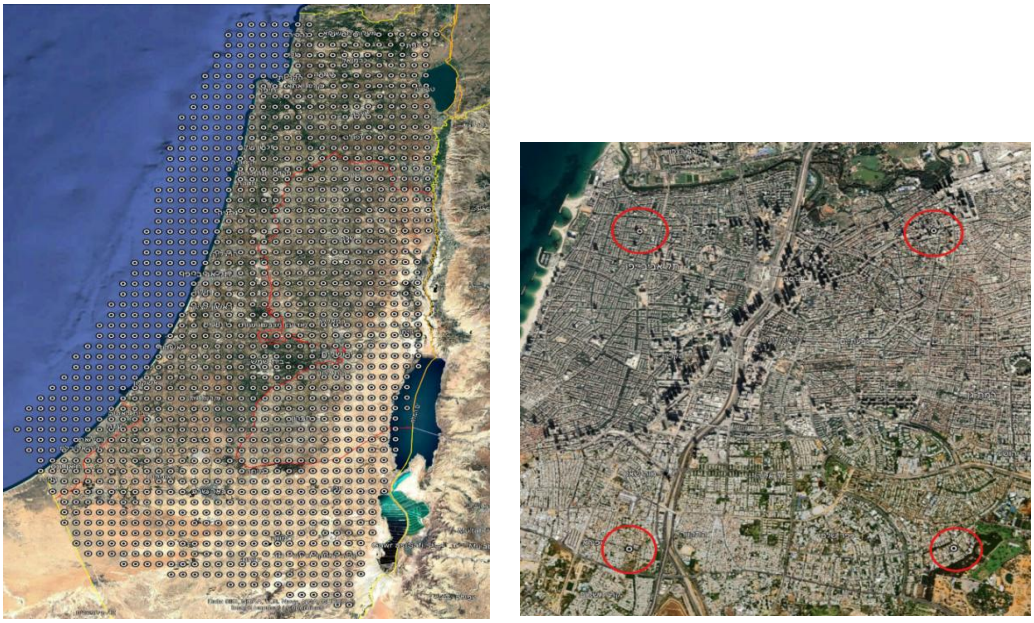
3.2.2 Filtering

We filtered out nighttime hours, as solar radiation prediction is not meaningful during those periods. Only daylight hours were retained for both graph construction and model training. Specifically, we restricted the data to the time range between 4:00 AM and 3:45 PM, ensuring the model focuses on periods with potential solar irradiance.

Due to computational constraints, we limited the spatial scope of our data to a specific region - namely, the area surrounding Jerusalem. This decision was made to reduce the size and complexity of the input graphs and enable efficient model training. To guide this selection, we considered the climatic diversity across Israel, which can be broadly divided into approximately three distinct climate zones. The Jerusalem area belongs to a specific climatic region characterized by its own typical patterns. Accordingly, we filtered the full dataset and retained only the PV sites located within this zone.

3.2.3 Graph Modeling

We structured the data as a graph, where each graph represents a specific time snapshot. The nodes in the graph correspond to PV sites - a total of 140 PV sites across Israel mainly nearby Jerusalem - arranged on a spatial grid with a resolution of 4×4 km, as shown in Figure 4. Each node represents a single PV site, characterized by its geographical coordinates (latitude and longitude) and associated GHI values. To capture spatial relationships between PV sites, we defined edges based on geographic proximity. An undirected edge is formed between two nodes if the Euclidean distance between them is less than or equal to 16 kilometers. This threshold was selected based on meteorological reasoning: the typical size of a cloud ranges from 10 to 15 kilometers. Thus, PV systems within this distance are likely to experience similar irradiance fluctuations due to shared cloud coverage.



(a) **(b)**

Figure 4. (a) The distribution of PV sites across Israel, where each site corresponds to a node in the graph. (b) Spatial resolution of 4×4 km.

Node features were selected based on a correlation analysis. We observed that each node exhibits a strong temporal correlation with its own past values, particularly within a window of up to five hours. Accordingly, we represented each node's features as a time series of its historical GHI values over the past two hours, sampled at 60-minute intervals. In addition, significant correlations were observed with values from previous days, leading us to include the 24, 48, and 72-hour lags as additional features. This resulted in a total of five input values per node, effectively capturing recent GHI dynamics while maintaining a manageable feature size for the model.

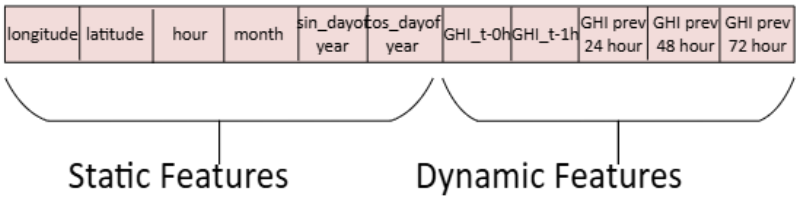


Figure 5. Node features.

This graph-based representation allows us to effectively model both the spatial distribution and the temporal dynamics of solar irradiance, serving as a suitable input structure for downstream GNN-based forecasting models.

3.3 Model's Architecture

To implement our spatiotemporal forecasting approach, we designed a hybrid deep learning model that combines LSTM units with GAT. The architecture reflects the structure described in the theoretical section and is tailored to capture temporal patterns within each PV site alongside spatial interactions across neighboring sites.

Each node in the graph corresponds to a PV location and processes its recent GHI time series using LSTM layers. The resulting temporal embeddings are then passed through a GAT layer, which enables dynamic, attention-based aggregation of information from spatially proximate nodes. This layered structure allows the model to effectively learn site-specific trends while accounting for shared influences such as cloud movement and localized weather phenomena.

The model's architecture represented in Figure 6:

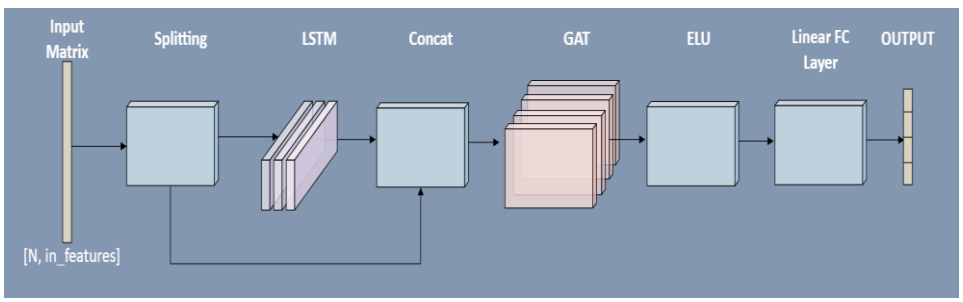


Figure 6. Model's Architecture.

3.4 Model Evaluation and Metrics

In order to evaluate how the model is performing we used two defined metrics commonly used in the literature, regarding PV power and GHI prediction problems [5].

The peak normalized root mean square error (NRMSE) is a commonly used metric which gives us an indication how well the model's predictions. The metric is shown Equation 8:

Equation 8: $$\boldsymbol{NRMSE(v,h)} = \sqrt{\frac{1}{T}\sum_{\tau \in S}\left(\frac{\hat{p}_v(\tau+h)-p_v(\tau+h)}{p_v^{max}}\right)^2}$$

where $p_v(\tau)$ and $\hat{p}_v(\tau)$ denote the ground truth GHI and predicted GHI, respectively, of PV v at time τ and horizon h. The p_v^{max} is the maximum GHI of site v over the evaluation period S. As described in section 3.2.1 our evaluation period covers January 5-17, 2017. In addition, T is the number of time steps in the evaluation interval.

From Equation 8, we can compute the NRMSE per PV:

Equation 9*:* $$\boldsymbol{NRMSE(h)} = \sqrt{\frac{1}{T\cdot n}\sum_{i=1}^{n}\sum_{\tau \in S}\left(\frac{\hat{p}_v(\tau+h)-p_v(\tau+h)}{p_v^{max}}\right)^2}$$

Where n is the number of the PV sites in our grid.

The second metric used is the average normalized mean absolute error (NMAE) defined as:

Equation 10: $$\boldsymbol{NMAE(v,h)} = \frac{\sum_{\tau \in S}|\hat{p}_v(\tau+h)-p_v(\tau+h)|}{\sum_{\tau \in S} p_v(\tau+h)}$$

The notations are the same as for the NRMSE metric. From Equation 10 we can compute the NRMA metric for a specific time horizon:

Equation 11: $$\boldsymbol{NMAE(h)} = \frac{1}{n}\sum_{i=1}^{n}\frac{\sum_{\tau \in S}|\hat{p}_v(\tau+h)-p_v(\tau+h)|}{\sum_{\tau \in S} p_v(\tau+h)}$$

After training the model, we used the test set to evaluate the model's performance on unseen data.

4 Analysis of Results

4.1 Correlations

As detailed in Section 3.2.3, we began by computing the autocorrelation of the GHI time series within each individual node. To obtain a global view of temporal dependencies across the entire network, we then averaged the autocorrelation values across all nodes for each time lag. The following figure presents the resulting average autocorrelation over 96 time lags, corresponding to a 24-hour period sampled at 15-minute intervals. This analysis provides insight into the periodicity and temporal structure inherent in the irradiance data.

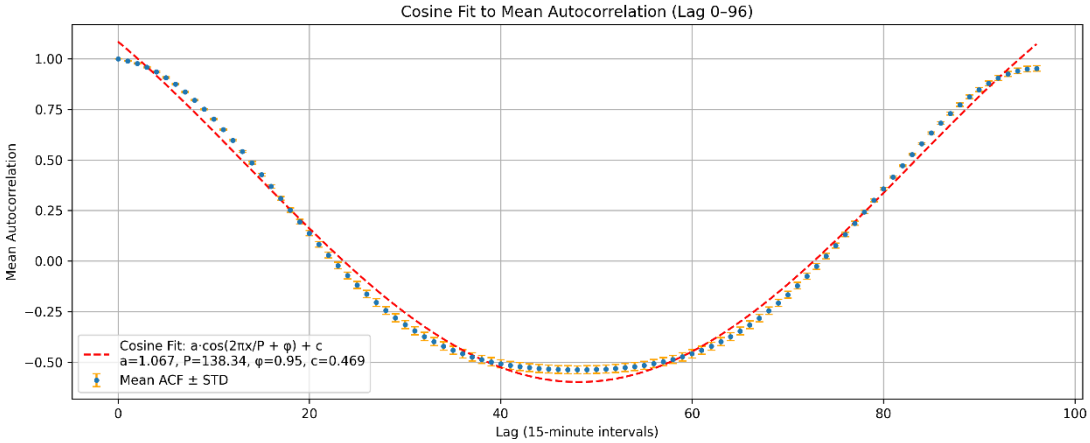


Figure 7. Average autocorrelations for 24 hours.

In addition, to analyze the seasonal trends in GHI across the year, we computed the autocorrelation of GHI over a full year for each PV node. We then averaged the autocorrelation values across all nodes for each lag, resulting in a single representative autocorrelation vector spanning the entire year.

To extract a single trend value per day, we divided the year into daily segments of 96 lags (corresponding to 15-minute intervals over 24 hours). For each day, we identified the lag with the maximum positive autocorrelation value, reflecting the dominant temporal dependency for that day. The resulting time series captures the evolution of GHI temporal consistency throughout the year.

Figure 8 presents the resulting trend, where the shaded region around the curve represents variability (e.g. standard deviation).

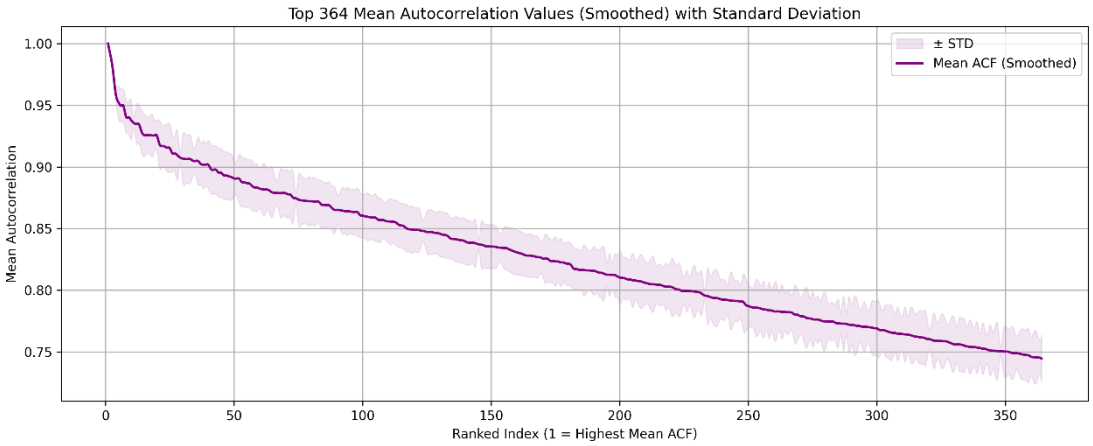


Figure 8. Average autocorrelation across one year.

4.2 NRMSE and NMAE

As described in section 3.4 we calculated the NRMSE as is Equation 9, and the NMAE as is Equation 11. The results are depicted in Table 1:

Time Horizon [min]	NRMSE [%]	NMAE [%]
15	8.25	11.35
30	9.13	13.16
45	10.02	15.14
60	10.76	16.46

Table 1. Our project's metrics results.

Table 2 presents a quantitative comparison of our model's performance and other literature's models in terms of NRMSE across multiple forecasting horizons.

Time Horizon [min]	15	30	45	60
Our model	8.25	9.13	10.02	10.76
TSM-GAT[1]	4.75	6.25	7.25	8.0
SimVPV2	6.6	7.8	8.6	9.2
PredRNN++	5.3	7.2	8.3	9.1

Table 2. Model's NRMSE results compared to literture.

[1] The values for TSM-GAT (multi-windows graph attention network) were estimated from performance plots provided in [5].

Table 3 presents a quantitative comparison of our model's performance and other literature's model[2] in terms of NMAE across multiple forecasting horizons:

Time Horizon [min]	15	30	45	60
Our model	11.35	13.16	15.14	16.46
TSM-GAT	5	8.15	10.45	13

Table 3. Model's NMAE results compared to literture.

4.3 Loss Graphs

Figure 9 presents the loss curves for both the training set and the test set. As shown, the training loss continues to decrease steadily across epochs, while the test loss stabilizes at a higher value. This gap between the curves indicates overfitting: the model fits the training data increasingly well, but its ability to generalize to unseen test data is limited. The reasons for this behavior, including dataset limitations and computational constraints, are discussed further in section 5.

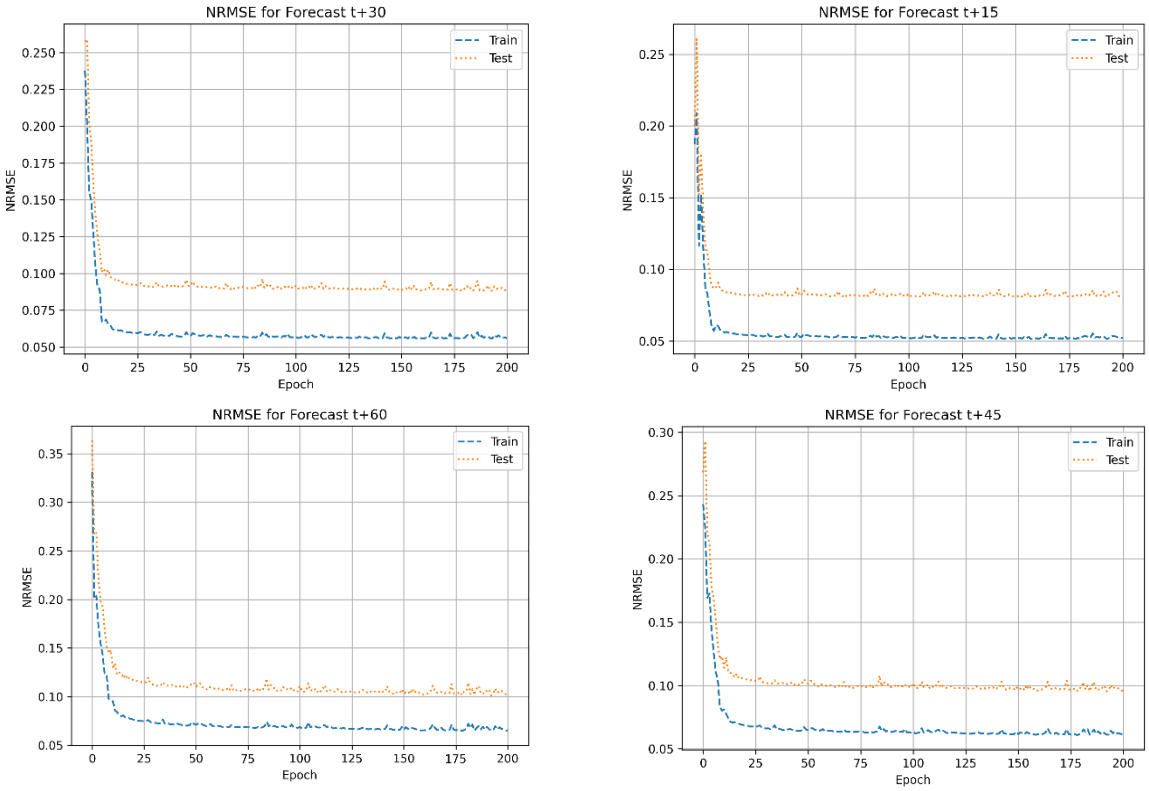


Figure 9. Loss graphs.

[2] NMAE metric was calculated only in the TSM-GAT model.

4.4 Prediction Versus Ground True

Figure 10 illustrates the true ground-truth GHI signal compared to the predictions generated by the LSTM-GAT model for two forecasting horizons: 15 minutes and 60 minutes ahead.

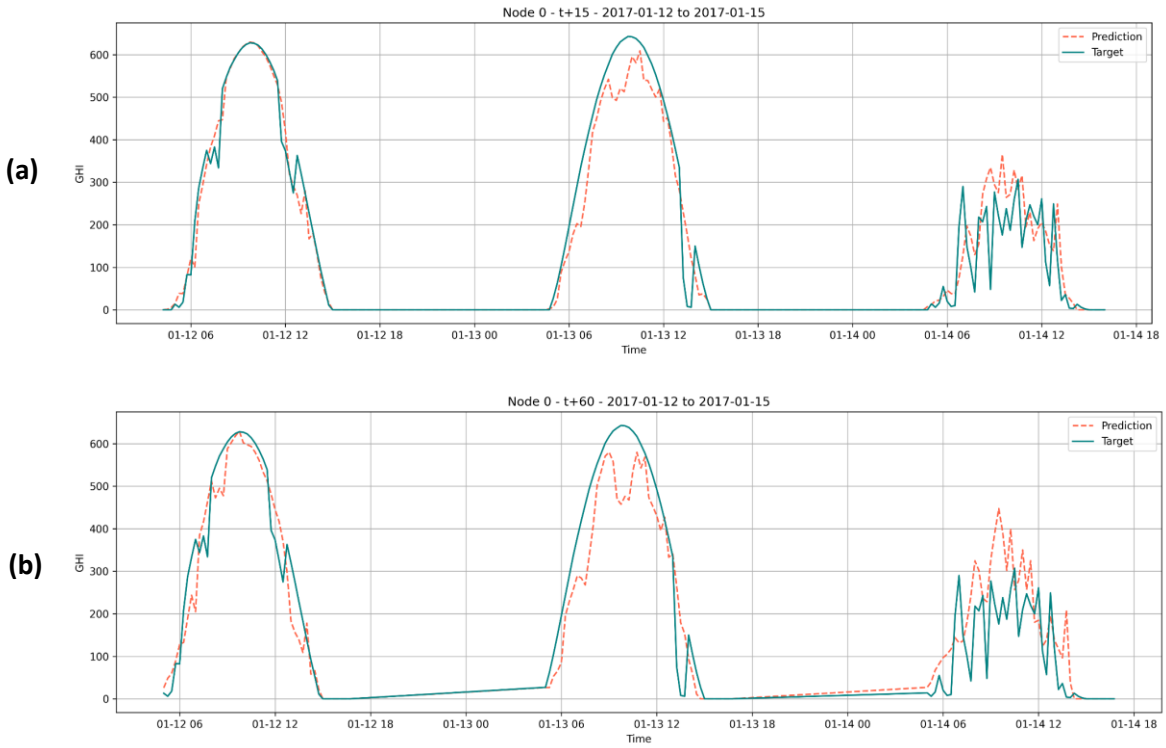


Figure 10. (a) a specific node prediction time series for 15 minutes prediction ahead. (b) 60 minutes prediction of the model at the same node.

5 Conclusions and Further Work

Our project sets out to improve short-term forecasting of GHI by developing a spatiotemporal deep learning model that integrates LSTM units with GATs. The goal was to predict GHI at multiple forecasting horizons - 15, 30, 45, and 60 minutes - across a network of PV sites in Israel.

5.1 Summary of Results and Model Behavior

The results achieved demonstrate that our model successfully captures both the temporal dynamics and spatial dependencies inherent in GHI data. As shown in Tables 1-3, the model achieved reasonable accuracy, with NRMSE values ranging from 8.25% to 10.76% and NMAE values from 11.35% to 16.46% across all forecasting horizons. While these results do not outperform all existing models in the literature - particularly TSM-GAT - they are promising given the computational constraints and the simulated nature of our dataset.

The hybrid architecture, combining LSTM and GAT layers, proved effective for the problem structure. It learns temporal sequences at each PV site while also leveraging spatial correlations between neighboring sites. This is evident in Figure 10, where the model closely follows the ground truth signal. On clear-sky days with stable irradiance, predictions align well with actual GHI values. On noisier days, characterized by stronger fluctuations, the model still captures the overall trend, albeit with reduced precision—highlighting directions for improvement, as further discussed in Section 5.2.

The loss curves (Figure 9) reveal signs of overfitting. This stems mainly from computational constraints: training was limited to shorter, non-continuous time windows, whereas the test set was kept as continuous segments to allow fair comparison with prior studies. Notably, when the test set was restricted to the same time windows used for training, no overfitting was observed and the training and test loss curves converged to similar values. These limitations, and their implications for model performance, are discussed in greater depth in Section 5.2.

In contrast, the prediction-versus-ground-truth graphs (Figure 10) suggest that the model generalizes reasonably well to unseen data. As expected, and consistent with prior work, forecasting accuracy declines as the prediction horizon increases. The model performs better at the 15-minute horizon than at 60 minutes ahead, across both evaluation metrics (NRMSE

and NMAE), since shorter horizons are naturally less sensitive to the compounding uncertainty of atmospheric variability.

5.2 Suggestions to Improve Model Performance

Despite the progress made, several opportunities exist to improve the model's performance:

- **Expanded graph modeling and training volume under computational constraints:** Due to limited computational resources, we faced a trade-off between the spatial size of each input graph (i.e. the number of PV nodes included per snapshot) and the number of training examples we could process. Larger graphs provide richer spatial information and improve the model's ability to learn complex inter-site dependencies, but they require significantly more memory and computational power. To keep training feasible, we limited the geographic scope of each graph - focusing on the Jerusalem region - and reduced the number of examples used per epoch. With access to more powerful hardware, it would be possible to include more nodes per graph and increase the number of training samples, thus enabling the model to learn from both broader spatial relationships and a more diverse temporal dataset. This would likely result in improved generalization and better performance, especially under dynamic weather conditions.
- **Limitations of the dataset and model reliance on simulated GHI values:** One notable limitation of our current approach lies in the use of simulated GHI data, provided by NREL, rather than real-world sensor measurements. While this dataset enabled experimentation over a large, structured spatial grid, it may not fully reflect the noise characteristics, inconsistencies, or sensor errors present in real environments. This gap could impact the model's generalizability when applied to actual deployment scenarios. To improve realism and robustness, future iterations of the model should incorporate real-world PV sensor data where available. Additionally, combining simulated and empirical data sources could help bridge the domain gap and better train the model under varying conditions.
- **Geographical scaling**: Expanding the model beyond the Jerusalem area to include multiple climate zones in Israel would increase generalizability. This was our first goal however it requires more computing power that we didn't have.
- **Model tuning**: Further hyperparameter optimization, additional attention heads in the GAT layer, or experimenting with alternative spatiotemporal architectures (e.g., Transformers, Graph Convolutional Networks) could lead to improved results.

5.3 Direction for Future Research

5.3.1 Cloud-based intermediate forecasting as an alternative strategy

Interesting direction for future research involves shifting the focus from direct GHI forecasting based solely on historical irradiance values to a cascaded approach that models cloud dynamics explicitly. Since cloud cover is the primary source of short-term variability in solar irradiance [8], predicting cloud movement - using tools such as satellite imagery or cloud simulators - may serve as an effective proxy for downstream GHI estimation and, eventually, PV power forecasting.

During the project, we made a preliminary attempt to explore this idea by integrating a cloud simulation tool (available in our project repository). However, the tool required high-resolution image input that matched our 4×4 km spatial grid, leading to substantial computational overhead. These resource constraints limited our ability to pursue this direction further. Nonetheless, the potential of cloud-based intermediate modeling remains high, and further exploration could lead to enhanced forecasting performance.

5.3.2 Impact of PV spatial density on forecast accuracy

An important research direction is to systematically investigate how the spatial density of PV sites influences model performance. In graph-based models, denser spatial configurations provide richer connectivity and more meaningful neighborhood contexts. Conversely, sparse PV deployment results in fewer edges and reduced spatial context, which may degrade forecasting accuracy. Studying this relationship could help identify optimal sensor deployment strategies or guide graph construction in future implementation.

5.3.3 Inference at unsensed locations via spatial interpolation

Building on the previous point, the model could potentially be extended to predict GHI at locations that are not instrumented with PV sensors. By leveraging spatial correlations learned from neighboring nodes, the model could estimate irradiance in unsensed regions. This capability would be especially valuable in scenarios where sensor deployment is limited by cost, terrain, or infrastructure, and could significantly enhance solar resource mapping and short-term planning.

6 Project Documentation

This section provides an overview of the materials and resources developed during the project. The full documentation is available online and includes software code, datasets, simulation tools, and user guidelines.

6.1 Documentation Location

The full documentation and project repository can be found at the GITHUB repository link: https://github.com/galrtz/SolarMLProject

6.2 Description of the Project Documentation

The documentation includes the following components:

- **Source Code** - Full implementation of the GAT-LSTM forecasting model, including training scripts, evaluation routines, and visualization tools.
- **Dataset Preparation Tools** - Scripts used to process and filter the NREL GHI data, generate graph representations of the PV network, and structure the time series inputs.
- **Simulations and Plots** - Visualizations of autocorrelation analysis, prediction vs. ground truth graphs, loss curves, and other experimental results.

6.3 Description of the Project Files

Folder Name	Folder Description	Files	File Description
Clouds_Simulator	Clouds simulator relevant scripts		
Correlations	Correlation calculation scripts		
Data	Processed datasets for training and testing	2018_train_data.zip	GHI training dataset for year 2018
		2019_train_data.zip	GHI training dataset for year 2019
		2020_train_data.zip	GHI training dataset for year 2020
		2021_train_data.zip	GHI training dataset for year 2021

		test_set_directory.zip	Test set based on 2017 data for model evaluation
Metrics	Evaluation scripts for error metrics and comparisons	Actual_vs_prediction.py	Generates comparison plots between predicted GHI and ground truth
		actual_vs_pred_one_pv.py	Produces prediction vs. actual GHI plot for a single selected PV site
		calcualating_nmrse_nmae.py	Computes NRMSE and NMAE metrics
Model	GNN model definitions and training logic	model_with_node_splitting_in_train.py	Model implementation that splits nodes during training to optimize memory usage and batch efficiency
		model_without_node_splitting_in_train.py	Full-graph training implementation, using all nodes without splitting - suitable for smaller graph sizes
Project	Project documentation	Final_Presentation.pptx	
		Poster_3022.pptx	
		Project_Book_3022.pdf	
Results	Forecast output plots and visualizations. Visualized predictions for 15, 30, 45, and 60-minute horizons	+15	15, 30, 45, 60 minutes prediction versus actual GHI png files
		+30	
		+45	
		+60	
data_proccessing	Scripts to create and convert datasets into graph format	creating_data_set.py	Generates the dataset to be used as input for splitting script
		creating_graphs.py	Builds spatial graphs of PV sites
		peventong_data_leackge.py	Filters the existing data set to prevent data leakage in training
		splitting.py	Splits the prepared dataset into multiple time-based subsets, each used to generate a separate graph

7 References

[1] Alet PJ, Efthymiou V, Graditi G, Henze N, Juel M, Moser D, Nemac F, Pierro M, Rikos E, Tselepis S, Yang G. Forecasting and observability: Critical technologies for system operations with high PV penetration. In: 32nd European photovoltaic solar energy conference and exhibition. 2016.

[2] Ajith, M., & Martínez-Ramón, M. (2023). Deep learning algorithms for very short term solar irradiance forecasting: A survey. Renewable and Sustainable Energy Reviews, 182, 113362. https://doi.org/10.1016/j.rser.2023.113362

[3] Zhou, H., Zhang, Y., Yang, L., Liu, Q., Yan, K., & Du, Y. (2019). Short-term photovoltaic power forecasting based on long short term memory neural network and attention mechanism. *IEEE Access*, *7*, 78063–78074. https://doi.org/10.1109/ACCESS.2019.2923006

[4] Ren Xiaoying, Gao Yanning, Zhang Fei, Gao Lu. A deep learning-based method for ultra-short-term PV power prediction. J Phys Conf Ser 2022;2260(1):012056. http://dx.doi.org/10.1088/1742-6596/2260/1/012056.

[5] Simeunović, J., Schubnel, B., Alet, P.-J., Carrillo, R. E., & Frossard, P. (2022). Interpretable temporal-spatial graph attention network for multi-site PV power forecasting. *Applied Energy*, *327*, 120127. https://doi.org/10.1016/j.apenergy.2022.120127

[6] Salim, M. S., Najim, J. M., & Salih, S. M. (2013). Practical evaluation of solar irradiance effect on pv performance. *Energy Science and Technology*, *6*(2), 36–40. https://doi.org/10.3968/j.est.1923847920130602.2671

[7] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). *Graph attention networks* (No. arXiv:1710.10903). arXiv. https://doi.org/10.48550/arXiv.1710.10903

[8] Inman, R. H., Pedro, H. T. C., & Coimbra, C. F. M. (2013). Solar forecasting methods for renewable energy integration. *Progress in Energy and Combustion Science*, *39*(6), 535–576. https://doi.org/10.1016/j.pecs.2013.06.002