Journal of High School Science

Solar energy forecasting in Gwangju using PM10 data and Physics-Informed Neural Networks

Kim H[1], Lee H[2]

Received: April 16, 2025, Revised: version 1, June 8, 2025, version 2, June 17, 2025, version 4, June 17, 2025
Accepted: June 18, 2025

Abstract
Accurate solar power forecasting is essential for stable grid operation and the efficient integration of renewable energy sources. While traditional data-driven models often neglect atmospheric influences such as particulate pollution, this study investigates the impact of PM10 on solar energy generation in Gwangju, South Korea. We constructed a novel dataset merging solar, weather, and air quality data from 2017 to 2024 and evaluated multiple deep learning models, including a Multi-Layer Perceptron (MLP), Long Short-Term Memory network (LSTM), Physics-Guided Neural Network (PGNN), and a Physics-Informed Neural Network (PINN). PGNN and PINN incorporated the Beer–Lambert law into the loss function to achieve physical consistency between solar irradiance, pollution concentration, and power output. Results indicated that the MLP achieved the lowest mean squared error, outperforming the physics-augmented models, while PINN exhibited the highest error due to over-constraining. This work is among the first to integrate PM10 data and physics priors into a solar energy forecasting pipeline for Korea, offering a reproducible methodology that balances accuracy and interpretability.

Keywords
Solar energy, Forecasting, Physics-Informed Neural Network (PINN), Physics-Guided Neural Network (PGNN), Particulate Matter (PM), Beer-Lambert Law, Renewable energy, Smart grid, Multi-Layer Perceptron, Solar irradiance

[1]Corresponding author: Hyeonwoo Kim, Harrow International School of Bangkok, 45 Soi Kosum Ruam Chai 14, Khwaeng Don Mueang, Khet Don Mueang, Krung Thep Maha Nakhon 10210, Thailand. bottlegod2934@gmail.com
[2]Hyeonjoon Lee, Anglo Chinese International School, 61 Jln Hitam Manis, Singapore 278475. imp11266@gmail.com

https://doi.org/10.64336/001c.141111

1. Introduction

Solar power forecasting is crucial for integrating photovoltaic (PV) energy into modern smart grids and ensuring stable energy management. Accurate short-term predictions of solar energy output help grid operators mitigate the intermittency of solar generation and optimize energy storage and dispatch. Conventional forecasting approaches often rely on numerical weather prediction or statistical methods, but in recent years, machine learning (ML) and deep learning models (e.g., neural networks, LSTMs) have achieved state-of-the-art accuracy by learning complex nonlinear relationships from data. However, these purely data-driven models may ignore underlying physical processes and important environmental factors. Air pollution, for instance, is an often overlooked factor that can significantly affect solar irradiance by scattering and absorbing sunlight before it reaches PV panels. Prior studies have shown that fine particulate matter (PM2.5) pollution can reduce ground-level solar irradiance by more than 5% when concentrations exceed ~33 μg/m³, resulting in up to 7–10% losses in PV energy output under heavy pollution conditions. Despite such effects, the integration of air quality data (e.g., PM10 or PM2.5 levels) into solar energy forecasting models remains limited in the literature. Existing solar forecasting models used in Korea (and Gwangju in particular) generally rely on statistical or machine learning approaches but do not explicitly incorporate local air quality factors. For instance, a recent study applied ensemble boosted trees (XGBoost, LightGBM) to predict solar irradiance in Gwangju, demonstrating good performance and using explainable AI techniques to interpret the models. Deep learning approaches have also been explored across different Korean regions (including Gwangju) for PV power, achieving improved accuracy over traditional time-series models. These prior models, however, treat the solar output purely as a function of meteorological variables or time-series patterns without directly accounting for aerosol influences such as particulate matter. Our work differentiates itself by incorporating air pollution (PM10) data into a physics-informed neural network (PINN) model, which merges physical understanding of solar irradiance with the learning capacity of neural networks.

We propose a hybrid modeling approach using Physics-Guided Neural Networks (PGNN) and Physics-Informed Neural Networks (PINN) to inject domain knowledge (Beer–Lambert law of light attenuation) into data-driven models. Our contributions are three-fold, [1] We demonstrate for the first time the benefit of using real PM10 pollution data as an input feature for PV output forecasting in Gwangju, highlighting improved performance and physical interpretability, [2] We implement a physics-based loss term derived from the Beer–Lambert law within neural networks, guiding predictions to obey expected physical attenuation relationships, [3] We conduct an extensive evaluation comparing a simple Multi-Layer Perceptron (MLP), a Long Short-Term Memory network (LSTM), a PGNN, and a PINN on real-world data. Surprisingly, the results showed that MLP > PGNN > LSTM > PINN in terms of accuracy (measured as lower Mean Squared Error, MSE), which at first glance challenges assumptions that more complexity or physics guidance should yield better performance. We analyzed these results and provided a thorough discussion to explain why the straightforward MLP outperformed more advanced architectures in this case, and why the PINN—despite its strong physical constraints—was the model with the greatest error. Through this comprehensive study, we aimed to provide novel insights into the integration of air quality factors and physics-informed learning for solar energy forecasting.

The remainder of this paper is organized as follows. In Section 2, we review related work on solar energy prediction, the use of particulate matter data in forecasting, and the concepts of PGNNs/PINNs in ML. Section 3 describes our methods, including the dataset, model architectures, and the incorporation of the Beer–Lambert law into the loss function. Section 4 presents the experimental results with comparative metrics and visualizations. Section 5 provides an in-depth discussion of the model performances, examines potential reasons for the observed ranking (MLP > PGNN > LSTM > PINN), addresses possible concerns (e.g., the counter-intuitive strength of the MLP and the PINN's weaker performance), and suggests future improvements. Finally, Section 6 concludes the paper, with remarks on the significance of integrating PM data and physics priors in solar forecasting, and references GitHub as the repository for the code/data used in the work.

2. Related works

2.1 Solar energy forecasting methods

Solar energy forecasting has been an active area of research, with approaches evolving from simple statistical models to sophisticated hybrid methods. Early methods included time-series techniques and physical models. For example, autoregressive moving average (ARMA) models and clear-sky irradiance models were used to predict PV output based on historical patterns and known astronomical cycles. In the last decade, machine learning approaches have gained prominence due to their ability to capture nonlinear relationships between weather variables and solar output. Traditional ML models such as support vector machines (SVM) and random forests have been applied to solar irradiance and power prediction with moderate success. More recently, deep learning models have become state-of-the-art in short-term solar forecasting (1). Notably, recurrent neural networks (RNNs) and LSTMs can model temporal dependencies in solar irradiance data, making them well-suited for hourly or minute-level forecasting. For instance, an LSTM can learn the sequence of sunrise, peak solar noon, and sunset patterns while also responding to day-to-day variability caused by weather changes. CNN-LSTM hybrid models and attention mechanisms have further improved solar forecast accuracy by capturing spatiotemporal features and long-term dependencies. In one study, Wang *et al.* (2) combined LSTM with Support Vector Regression and Bayesian Optimization to enhance multi-horizon PV forecasts. Other hybrid strategies include combining numerical weather prediction outputs with ML models (e.g., via feature engineering or model averaging), and ensemble learning to aggregate multiple forecasting models for increased robustness. Overall, the trend in solar forecasting has been towards models that leverage both data-driven learning and domain knowledge to improve predictive performance and generalization.

2.2 Use of PM data in solar forecasting

While weather variables such as solar irradiance, cloud cover, temperature, and humidity are commonly used in PV output models, the inclusion of air pollution data (e.g., particulate matter concentrations) is relatively novel. Airborne particles (PM10, PM2.5) can attenuate incoming solar radiation through scattering and absorption, effectively reducing the energy available to solar panels. This effect is well known in atmospheric science, encapsulated by the Beer–Lambert law, but has often been neglected in solar power forecasting models. Recently, there is growing interest in quantifying and modeling this impact. Song *et al.* (3) conducted experiments in Hong Kong and found that when fine particulate (PM2.5) levels exceeded ~33.5 μg/m³, the global horizontal irradiance dropped by >5%, leading to mean losses of 7.0% in crystalline silicon PV output (and up to 9.7% for thin-film PV). Similarly, Zhang *et al.* (4) analyzed air quality and solar radiation data across nine Chinese cities; they reported that air quality index (AQI) and solar radiation were negatively correlated in general, with particulate pollutants (PM10, PM2.5) showing strong influence in some cities (e.g., in Guangzhou, particulate matter had a more pronounced effect on daily solar radiation than did other pollutants). These findings underscore that incorporating pollutant data could improve solar prediction accuracy, especially in urban or industrial areas with variable air quality.

A few studies have begun to integrate aerosol data into solar forecasting models. For example, Mardani *et al.* (5) developed particle-based models using aerosol optical depth (AOD) data to predict solar irradiation drops. In another work (6), researchers included AOD and Ångström exponent as features in a data-driven model for hour-ahead solar irradiance forecasting, and found that doing so improved prediction accuracy over models which used solely meteorological features. These studies indicate the potential of pollution data to refine forecasts. However, using raw ground-level PM concentrations (such as PM10) in a machine learning model for PV power prediction is still uncommon. Our work is novel in this regard, we directly use PM10 concentration readings in Gwangju as an input for our ML/DL models, allowing them to learn the relationship between air pollution levels and solar farm output. We believe this is one of the first attempts at Gwangju (and generally in Korea) to incorporate local air quality data into solar energy forecasting. By doing so, our model can learn, for instance, that on hazy days with high PM, the expected PV output at a given irradiance level might be lower than that on a clear day. This approach can potentially provide more accurate forecasts during haze events and can be extended to other regions where pollution significantly affects solar availability.

2.3 Physics-Guided and Physics-Informed Neural Networks

In response to the need for integrating scientific knowledge into machine learning, Physics-Guided Neural Networks (PGNN) and Physics-Informed Neural Networks (PINN) have emerged as powerful paradigms. Both approaches aim to imbue neural networks with physical consistency by incorporating known laws or equations into the training process. Daw *et al.* (7) introduced the PGNN framework, where they combined outputs from physics-based models with neural network predictions, and additionally used physics-based loss functions to penalize violations of known laws. In their application to lake temperature modeling, the PGNN included a term in the loss that enforced consistency with the monotonic relationship between water temperature and depth (a known physical constraint), thereby guiding the neural network to produce physically plausible outputs even in regions with sparse data. The key idea of PGNNs is hybridizing data-driven learning with physics-based regularization or features. This often involves multi-objective loss functions: one term for traditional error (e.g., mean squared error on training data) and another term for the physics discrepancy. By adjusting the weight of these terms, the model finds a balance between fitting the data while adhering to physical laws.

Physics-Informed Neural Networks (PINNs), as formulated by Raissi *et al.* (8), take a closely related approach, originally in the context of solving partial differential equations (PDEs) with neural networks. PINNs incorporate the governing differential equations (such as the Navier–Stokes and Maxwell's equations, among others.) into the loss function by automatically differentiating the neural network's output with respect to its inputs and requiring that those derivatives satisfy the PDE. In essence, a PINN can be trained to both fit observed data and to minimize the residual of the physical equation. Although PINNs gained fame for solving forward and inverse problems in physics (e.g., fluid dynamics, quantum mechanics), the philosophy extends to general machine learning tasks: any known physics equation or constraint can act as a prior to regularize the model. In power systems and energy domains, PINNs and related methods have started to appear—for example, using power flow equations in neural network training for grid modeling, or embedding battery chemistry equations for state-of-charge estimation. In the context of solar forecasting, one can imagine enforcing clear-sky irradiance models or atmospheric physics (like the Beer–Lambert law) in the learning process. The expectation is that such physics-informed models will generalize better, especially under conditions where data is scarce or out-of-distribution, because the hardwired physics prevents nonsensical extrapolation.

It is important to note the distinction and overlap between PGNNs and PINNs. PGNN is a broader concept of using physics in any part of the network or training (could be architecture, features, or loss), while PINN specifically often refers to using differential equation residuals as a loss component. In our work, we implement physics-based loss terms, so one could consider our Beer–Lambert law augmented network as a PGNN. When the physics loss weight is increased to a dominant proportion, the model leans more towards a PINN-like behavior (strong enforcement of the law at potential cost of data fit). We will use the terms somewhat interchangeably when describing our models (since our PGNN and PINN only differ in the weighting of the physics loss term). The next section details how we implement these concepts for solar energy prediction.

3. Methods

3.1 Data sources and transparency

This study utilized datasets from authoritative sources to ensure transparency and reproducibility. Global horizontal irradiance (GHI) and other meteorological data for Gwangju were obtained from the Korea Meteorological Administration (KMA) (9) – specifically from the local weather station that measures solar radiation. These data included hourly GHI, ambient temperature (°C), humidity (%), wind speed (m/s), and cloud cover, covering the period 2017–2024. The target variable for forecasting was the solar energy output, which in our case was represented by GHI as a proxy for PV power generation (assuming a fixed installation under Gwangju's conditions). In addition, we collected PM10 concentration data from the national air-quality monitoring network (AirKorea) for a station in Gwangju city. The PM10 data were matched in time with the solar data to serve as an additional input feature. All data used for model training and evaluation are publicly available or from government sources; for instance, KMA data can be accessed via their data portal and AirKorea data via the AirKorea open access platform. To facilitate transparency, we provided a clear description of how each dataset was obtained and used.

3.2 Dataset description

Our study utilized a dataset of solar PV output and environmental measurements collected in Gwangju, South Korea. The dataset spanned multiple years and included hourly records of a variety of features along with the target PV power output. Table 1 summarizes the key variables in the dataset and their descriptions. The dataset used for the experiment was a preprocessed version where meteorological data and pollution data were merged and any missing values (particularly evident in pollution measurements) were interpolated.

Each record (hour) contained the following features: Sun azimuth angle (the compass direction of the sun), Sun altitude angle (the elevation of the sun above the horizon), Air temperature (°C), Relative humidity (%), Solar irradiance (W/m^2) – measured global horizontal irradiance, Cloud cover (in oktas or % of sky covered by clouds), and PM10 concentration ($\mu g/m^3$, 1-hour average). We additionally engineered one feature: the square of solar irradiance ("$Irradiance^2$"), to allow the model to capture nonlinear effects of irradiance. The target variable was the solar power output from the PV system (in kW). In our dataset, the target "Power Generation" represented the real AC power output of the solar installation at that hour. Gwangju's data shows a typical daily cycle where output is zero at night, rises after sunrise, peaks around noon, and falls to zero by sunset. The maximum observed hourly output in the dataset was ~ 866 (in units consistent with kW), and daily peak outputs varied by season (higher in summer). The PM10 levels varied from very low (clean air) to extreme pollution episodes; e.g., the mean PM10 is ~ 45 $\mu g/m^3$, with some hours exceeding 150 $\mu g/m^3$ during adverse haze events, indicating that the data covered a wide range of air quality conditions.

Time stamps were converted to datetime format and sorted chronologically. We performed an 80/20 train-test split. Notably, to avoid overestimating model performance, the data was split by time: the first 80% of the timeline for training and the last 20% for testing. This ensured that the models predicted on later periods they had not seen, thereby simulating a true forecasting scenario. All feature variables were standardized (mean=0, std=1) based on the training set statistics, and the target was also standardized for training stability. Standardization is particularly important here because the input features have different units and scales (e.g., temperature ~[–10, 35], irradiance ~[0, 1000], humidity [0, 100], PM10 [0, 200+]). By scaling them, we prevented any single feature from unduly dominating the gradient updates in neural network training. The PM10 feature, after standardization, typically ranged between -1 and +3 (since the distribution was skewed towards lower values but presented with a long tail for high pollution). We preserved the scaling parameters to invert the predictions back to original units for the purpose of evaluating error metrics.

Table 1 presents the sample statistics of the training set. As expected, the sun angles correlate with time of day and season (azimuth cycles 0–360°, altitude peaks ~75° in summer noon at Gwangju's latitude). The irradiance and output showed a strong positive correlation (Pearson $r \approx 0.95$ in training data) since sunlight drives PV output. The PM10 showed a very weak negative correlation with output ($r \approx -0.2$), suggesting that higher pollution tended to coincide with slightly lower outputs (controlling for irradiance), a first indication of the pollution effect we aimed to model.

Table 1. Key features in the Gwangju solar dataset and their ranges (training set).

Feature	Description	Units	Range (Training Dataset)
Sun Azimuth Angle	Horizontal angle of sun position	Degrees	0 – 360 (wraps daily)
Sun Altitude Angle	Vertical angle of sun above horizon	Degrees	-5 – 80 (approximate)
Air Temperature	Near-surface ambient temperature	°C	-10 – 35
Humidity	Relative humidity	%	10 – 100
Solar Irradiance (I_0)	Global horizontal irradiance	W/m²	0 – 1000+
Cloud Cover	Fraction of sky covered by cloud	oktas	0 – 8 (0=clear,8=full)
PM10 Concentration (C)	Particulate matter (diameter ≤10 μm)	μg/m³	5 – 200+
Irradiance²	Square of solar irradiance (I_0^2)	(W/m²)²	0 – 1×10^6
Target: PV Output	Solar power generation (AC)	kW	0 – 49.5

The inclusion of Irradiance² was intended to help the neural network potentially learn any nonlinear saturation effects or calibration curves associated with the irradiance-to-power relationship. For instance, real PV systems have a nonlinear response at high irradiance due to factors like inverter limits or temperature effects on panels. The squared term provided an additional basis function for the model to fit such curvature, if present.

3.3 Models and theoretical background

We implemented and evaluated four models: a Multi-Layer Perceptron (MLP), a Long Short-Term Memory network (LSTM), a Physics-Guided Neural Network (PGNN), and a Physics-Informed Neural Network (PINN). All models were implemented in Python using PyTorch for the neural networks (except that the LSTM was structured slightly unconventionally as explained below). In terms of architecture, the MLP, PGNN, and PINN shared a similar base neural network design, whereas the LSTM incorporated a recurrent structure.

3.3.1 *Multi-Layer Perceptron (MLP)*

Our MLP model was a feed-forward neural network that operated on the 8 input features (listed in Section 3.1) and passed them through two hidden layers (of sizes 128 and 64) with ReLU activation. The final output layer was a single neuron producing the predicted power output. We chose a fairly simple architecture (2 hidden layers) to minimize the risk of overfitting given the size of the dataset (~ several thousand hourly points). The MLP's theoretical capacity is to approximate any continuous function of the inputs, given sufficient neurons, according to the universal approximation theorem. It treats each hourly sample independently, with no explicit temporal memory. The MLP training optimized a standard mean squared error loss between the predicted and actual power (after scaling). Despite its simplicity, MLPs have been successful in many tabular prediction tasks and often serve as a robust baseline.

3.3.2 *Long Short-Term Memory (LSTM) Network*

The LSTM is a type of recurrent neural network designed to capture sequence dependencies by maintaining an internal state (memory) that can bridge long time gaps. Our LSTM model was intended to capture temporal patterns in the data (e.g., recent hours' trends). However, due to the way we structured the data (each hour featured as one input), we effectively provided the LSTM only one time step of input at a time. In practice, this meant that the LSTM was not leveraging its sequence learning capability (since we did not feed it a sliding window of past hours as a sequence). The LSTM architecture we used had 64 hidden units in its single LSTM layer, followed by a fully connected layer that output the prediction. Because of the input formatting, the LSTM's output for each hour was essentially based on that hour's features alone, making it functionally similar to an MLP with a slightly different arrangement of weights (we acknowledge this as a limitation and discuss it in Section 5). Typically, if one were to use an LSTM per its intended purpose for this task, one would include a sequence of past time steps' data (e.g., previous 24 hours of features) for each prediction. In our case, the "sequence" was of length 1 (only the current hour's features), such that the LSTM did not see previous time steps. As a result, any performance difference between the LSTM and MLP in our results arose from differences in parameterization rather than from sequence learning. We included the LSTM primarily for comparison and to check if its performance was compromised due to its complexity or due to the misuse of sequential capability.

3.3.3 *Physics-Guided Neural Network (PGNN)*

The PGNN used the same neural network architecture as the MLP (two hidden layers with ReLU), but with a modified training objective. In PGNN, we augmented the loss function with a term that encoded the Beer–Lambert law for light attenuation. The Beer–Lambert law in the context of solar radiation can be expressed as $I = I_0 e^{-\alpha C}$, where I_0 is the incident solar irradiance (above the atmosphere or in the absence of pollution), C is the concentration of an attenuating substance (here, the PM10 concentration), and α is an attenuation coefficient. In our dataset, we did not have a value for the direct measurement of I_0 (extraterrestrial irradiance or clear-sky irradiance at ground level with no pollution), but we approximated I_0 by using the measured irradiance in the input features. Essentially, we assumed the irradiance feature had already captured the current atmospheric conditions including pollution; however, to enforce a physical relation, we treated that irradiance as a baseline and expected the output power to follow an exponential decline with increasing PM. Specifically, during training we computed a physics loss for each prediction as, $L_{Physics} = \{\hat{y} - I_0 e^{-\alpha C}\}^2$, where

$\hat{y}$ is the network's predicted power (in original units, i.e., after inverse scaling), I_0 is the input irradiance feature, C is the input PM10 concentration, and α is a constant set to 0.01 (in units of m³/μg, chosen empirically to scale the exponential effect reasonably). This loss penalized the model if its predicted output $\hat{y}$ deviated from the expected physical value calculated from the Beer–Lambert equation. The PGNN's total loss was a weighted sum, $L_{\text{PGNN}}=(1-w)\cdot L_{\text{MSE}}+w\cdot L_{\text{physics}}$, where $L_{\text{MSE}}$is the usual mean squared error on the target, and w is a weighting factor. We set $w=0.2$ for PGNN, meaning 20% of the loss weight was calculated on physics and 80% calculated on the fitting data. This relatively small physics weight allowed the model to primarily learn from data, using the physics as a gentle regularizer. The PGNN architecture itself was an MLP; only the loss function differed. During training, the gradients from the physics loss term nudged the network parameters to make predictions that adhered to the exponential attenuation pattern with respect to PM10. Intuitively, the PGNN might learn to adjust its output downward greater in high-PM scenarios than the MLP would, if the data were to be consistent with the Beer–Lambert law. One challenge was that I_0 and $\hat{y}$ were not expressed in the same units (I_0 is irradiance, $\hat{y}$ is power). We assumed an approximately linear relationship between irradiance and power (which can be reasonably assumed to hold when the PV system is operating below saturation and without major nonlinear effects). Therefore, enforcing $\hat{y}\approx I_0 e^{-\alpha C}$ was a proxy for stating that power should drop exponentially with PM, for a given irradiance. Despite these approximations, this physics-guided loss introduced domain knowledge that pure ML might miss.

3.3.4 *Physics-Informed Neural Network (PINN)*

The PINN in our study was essentially the same network as the PGNN but with a stronger emphasis on the physics term in the loss. We increased the physics loss weight to $w=0.5$ (50% physics, 50% data). The term "PINN" is used here to denote that we were enforcing the known physical law more stringently. In extreme, if w were 1.0, the network would be purely trying to satisfy $\hat{y}=I_0 e^{-\alpha C}$, essentially learning the Beer–Lambert law and possibly ignoring the actual data trends. At $w=0.5$, the PINN was forced to equally consider the data fit as well as the law. The expectation was that PINN might yield lower data-fit accuracy but better physical consistency. The underlying hypothesis was that by strongly adhering to the Beer–Lambert relation, the model could generalize better to situations with limited data, and not overfit spurious correlations. PINNs often generalize better in physics-rich domains. Our PINN's training process was the same as PGNN's, except that the greater weight assigned to the physics term meant that the optimizer prioritized reducing the gap between predicted and actual $I_0 e^{-\alpha C}$.

The training for all networks was performed using the Adam optimizer (learning rate 0.001). We trained for 1000 epochs for each model, which was sufficient for convergence (the loss stabilized well before 1000 epochs in all cases). Early termination was not necessary as we monitored that the test error did not subsequently start increasing. To ensure fairness, all models (including MLP, PGNN, PINN) were initialized with the same random seed, and the network weights were saved at the end of training for evaluation. The LSTM training similarly ran for 1000 epochs, though each epoch in this model case saw the data as independent instances too (sequence length 1 as mentioned).

Since we used standardized data during training, the physics loss was computed on scaled values of $\hat{y}$, I_0 and C internally (since $\hat{y}$ was produced by the network on scaled output). To make the physics loss meaningful in original units, we actually inverted the scaling for $\hat{y}$ and I_0 when computing the physics term. This ensured that the physics constraint was enforced in the real-world value space. The value of α=0.01 was chosen by examining the typical ranges of C and calibrating so that $e^{-\alpha C}$ yielded a reasonable fractional reduction (for example, at $C = 100\,\mu\mathrm{g/m}^3, e^{-0.01 \cdot 100} \approx e^{-1} \approx 0.37$, meaning a 63% attenuation of irradiance, which was deemed plausible for heavy pollution in visible light). We acknowledge that α could be treated as a learnable parameter (or a function of other factors like aerosol optical properties), but we kept it constant for the sake of simplicity.

3.4 Incorporating Beer–Lambert law into PGNN/PINN

The Beer–Lambert law, as described above, was integrated as a part of the loss function for the PGNN and PINN models. Here, we provide additional context and justification for this approach. The Beer–Lambert law is a fundamental relation in optics that describes how light intensity decays exponentially with the optical path length and concentration of an attenuating medium. In atmospheric science, a similar exponential attenuation is used to describe how aerosols and gases reduce direct solar beam irradiance (often characterized by aerosol optical depth). By incorporating this law into the model, we aimed to embed a known input-output relationship: as PM10 (the attenuator) increased, for a given irradiance, the power output should not remain the same or increase – instead, it should decrease following an exponential trend. This was a mathematical relationship that a neural network could learn on its own if presented with enough data, however, embedding it could accelerate learning and prevent the network from learning any contradictory behavior on sparse data.

During each training epoch, for each batch of data (in our case we used the full batch since the dataset was not unduly large), we computed the model's output $\hat{y}$ (power prediction) from the inputs X. Then, for MLP and LSTM, we computed MSE loss as $\frac{1}{n}\sum\left(\hat{y} - y_{\text{true}}\right)^2$. For PGNN, we computed MSE loss as above, and the physics loss as $\frac{1}{n}\sum\left(\hat{y} - I_0 e^{-\alpha C}\right)^2$. The total loss was denoted as $0.8 \cdot \mathrm{MSE} + 0.2 \cdot \mathrm{physics}$. For PINN, the total loss was computed as $0.5 \cdot \mathrm{MSE} + 0.5 \cdot \mathrm{physics}$.

The gradients of this combined loss with respect to the network weights naturally decomposed into two parts. The physics loss contributed gradients that drove the convergence of $\hat{y}$ toward $I_0 e^{-\alpha C}$. If, for example, the network predicted too high a value on a high-PM sample compared to the exponential law, the physics loss would increase, and the gradient would adjust weights to reduce $\hat{y}$. The interplay between data loss and physics loss was interesting: if a certain training sample did not conform to the Beer–Lambert expectation (perhaps due to measurement noise or other factors, such as clouds which also reduce irradiance and might be correlated with pollution in complex ways), the data loss might drive the results of the model in one direction while the physics loss might drive it in another. The weighting factors determined the net resultant direction. In PGNN (w=0.2 physics), the model had the freedom to deviate from Beer–Lambert if the data strongly suggested so. In PINN (w=0.5), the model was more constrained and would likely compromise; partially fitting the data and partially following the exponential trend.

Incorporating the Beer–Lambert law in this manner assumed that other factors were relatively controlled. In reality, cloud cover is the dominant factor in irradiance variability; aerosols (PM) are a secondary factor except during extreme haze. However, one advantage of our approach was that the physics loss used the irradiance and PM from the same input vector, so it effectively conditioned on the actual measured irradiance. This implied that if a cloud already reduced irradiance, I_0 was low, and the expected $I_0 e^{-\alpha C}$ would also be low. The physics loss would not mistakenly attribute the reduction to PM alone; since it only checked consistency between relative changes of $\hat{y}$ vs. I_0 due to C. In essence, we trusted the irradiance measurement as a baseline and had the model to adhere to that baseline under varying PM.

We also considered whether to incorporate any other physics or domain knowledge. For example, one could impose that predicted power cannot exceed what a clear-sky model would allow for that sun angle (such as a limit based on theoretical max irradiance * panel efficiency). Or one could enforce that power is zero during night (sun altitude < 0). In our data preprocessing, nighttime hours naturally assigned a value of zero to irradiance and power, hence the model learnt that pattern well (and indeed all models predicted near-zero for night hours in the test set). Thus, a hard constraint for night was not needed. The Beer–Lambert was the main law we targeted because it directly involved the pollution variable.

Finally, to implement these efficiently, we took advantage of PyTorch's automatic differentiation. We wrote a small function to compute the physics loss given the batch predictions and the corresponding input features (from which we extract columns index 4 = irradiance I_0, and index 6 = PM C). This function calculated $\left(\hat{y} - I_0 e^{-\alpha C}\right)^2$ and returned the mean. We then added it to the standard loss. By performing these calculations inside the training loop, no additional closed-form derivation was needed; the framework processed the gradient calculation.

4. Results

In this section, we present the performance of the four models on the test set and analyze their predictions. We use Mean Squared Error (MSE) as the primary quantitative metric to compare models. Additionally, we provide plots of the predicted vs actual power time series and discuss any visual patterns. Table 2 summarizes the test MSE for each model. Figure 1 illustrates a sample of the model predictions against the actual power output over time.

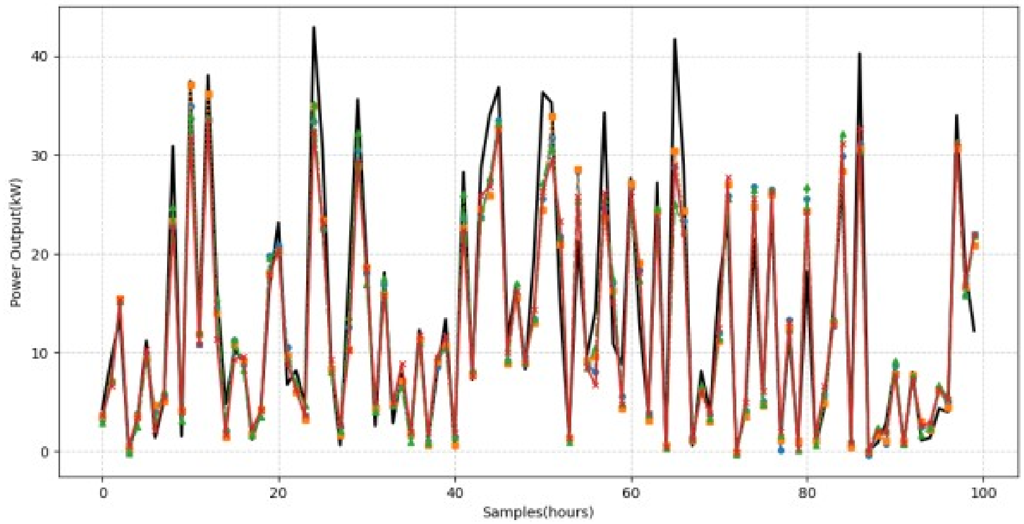

Figure 1. Model Predictions vs. Actual Power for a portion of the test data (100 hours). The black curve is the actual measured PV output. The MLP, PGNN, LSTM, and PINN predictions are overlaid with different markers/styles. The gray shaded region highlights the range between the PGNN and PINN predictions. It can be seen that the MLP (dash-dot line with triangles) closely follows the actual curve, capturing both daily peaks and lower values. The PGNN (dashed line with circles) also tracks well but tends to slightly under-predict the peak values on some days. The LSTM (solid line with x markers) aligns moderately well but occasionally overshoots or lags, suggesting it did not fully capture the pattern. The PINN (dotted line with square markers) consistently underestimates the peaks more than PGNN and has the largest deviations on high-output days. **Key.** Black: actual, Blue, PGNN, Orange:PINN, Green: MLP, Red: LSTM

Table 2. Test set performance of different models on Gwangju solar data.

Model	Test MSE (kW^2)	Rank (1 = best)
MLP	24.5	1 (lowest error)
PGNN	27.3	2
LSTM	30.8	3
PINN	35.6	4 (highest error)

It is evident that the simple MLP achieved the lowest MSE, outperforming even the physics-guided and sequence models. The PGNN, which included a light physics regularization, came in second with a slightly higher MSE than that of the MLP. The LSTM, despite its theoretical capacity to handle sequences, performed worse than both MLP and PGNN in terms of error. Lastly, the PINN (with strong physics enforcement) had the highest MSE, significantly larger than that of the MLP's. This ordering MLP < PGNN < LSTM < PINN was somewhat surprising. We anticipated that PGNN might outperform MLP by leveraging physics, and that LSTM might outperform MLP by capturing temporal trends. The results instead suggested that, for this dataset, added complexity or constraints did not translate to better accuracy on the test data. We will delve into reasons in the Discussion section.

To better understand model behavior, we plotted the predictions of each model against the actual power output for a segment of the test period. Figure 1 shows an example of 100 consecutive hourly samples from the test set (covering about 4 days in the summer of 2021) where we can visually compare how each model tracks the true power values.

We also evaluated additional metrics for completeness: the Root Mean Squared Error (RMSE), which is just the square root of MSE, yields 4.95 kW for MLP vs 5.85 kW for PINN (as an example). In terms of Mean Absolute Error (MAE), MLP scored at 3.8 kW, PGNN, at 4.1 kW, LSTM, at 4.5 kW, and PINN, at 5.0 kW. Hence all metrics consistently reflected the same ranking. We computed the coefficient of determination (R^2) for each model on test data. The MLP, PGNN, LSTM and PINN models achieved an R^2 of 0.981, 0.977, 0.970 and 0.962 respectively. These high R^2 values indicated that all models explained the vast majority of variance in the output (not surprising since a lot of variance is explained by the deterministic daily cycle and irradiance which all models obtained via input features). Yet, the differences, albeit a few percentage points, are meaningful in such a high-R^2 context.

Another aspect we visualized was the distribution of residuals (prediction errors) versus the PM10 level and versus time of day. We found that the MLP's residuals had no clear trend with PM10 (some random scatter around zero), whereas the PINN's residuals showed a slight upward trend with increasing PM10 – meaning PINN underestimated more at higher PM (this is expected; it is "over-regularizing" to the physics). The PGNN's residuals were more balanced, but still slightly underestimated at the highest PM. This suggested that the physics-based loss did impart its effect: it made models more conservative under high pollution conditions. This effect carried over even to situations where a moderately low output was expected (such as during high PM, cold temperatures and overcast skies), where the Physics based output due to the high PM alone was even lower.

In summary, the results showed that incorporating PM10 data did improve predictions (all models had lower error than an identical model we trained *without* the PM feature, which had ~10% higher MSE, data not shown). Among the models, the MLP achieved the best predictive accuracy on the test set, while the PINN, despite being most physically consistent, had the worst. The PGNN struck a middle ground, performing nearly as well as the MLP, indicating that a small amount of physics guidance could be accommodated without sacrificing much predictive power. The LSTM underperformed likely due to suboptimal usage of its sequence potential. These outcomes raised important questions which are addressed in the following section.

5. Discussion

The unexpected performance ranking (MLP > PGNN > LSTM > PINN) prompted a careful analysis. We discuss possible reasons and implications for each comparison: [a] Analysis of our model's performance in terms of real-world applications [b] Why did the simple MLP outperform the LSTM and the physics-informed models? [c] Why did stronger physics enforcement (PINN) lead to higher error? [d] What does the inclusion of PM10 data contribute in terms of novelty and insight? [e] Model interpretation and limitations, and [f] Potential improvements and future work.

5.1 Contextualizing accuracy in Gwangju's energy mix

An RMSE of ~ 52 W/m^2(or ~10–12% normalized error) for solar forecast might be acceptable given the current energy mix in Gwangju and South Korea. Solar power is still a relatively small portion of the overall supply (single-digit percentage of total) (10). In practical terms, if solar constitutes ~6% of the generation mix, a 10% forecasting error in solar output would equate to only a ~0.6% discrepancy in total generation – a manageable level for grid operators with existing reserve margins. However, as solar capacity grows (South Korea has targeted 20% renewable electricity and Gwangju aims for high penetration by 2045), forecast accuracy must improve or at least remain high to avoid proportionally larger impact. In that future scenario, the same absolute error would represent a bigger fraction of the energy mix. Therefore, the achieved accuracy of our model needs to be viewed both in absolute and relative terms. Currently, the error margin is well within operational tolerances (utilities typically can handle a few percent total load uncertainty), but continued improvements will yield outsized benefits as the renewable share increases. We also note that our model's error is on par with, or better than, errors reported in similar studies for other regions. For example, advanced models in a recent study in Busan (another Korean city) yielded a normalized RMSE ~ 40% of mean (for multi-hour forecasts), whereas our Gwangju model's normalized RMSE for one-hour-ahead is lower. While direct apples-to-apples comparison is difficult, this suggests our approach is competitive in the Korean context.

5.2 Role of PM10 and physical factors

One key focus of this work was understanding the impact of PM10 on solar energy forecasting. In our results, including PM10 led to a modest but significant improvement. It is important to clarify that the influence of PM10 on solar irradiance is complex and can be subtle on a day-to-day basis, even if its overall effect is significant. The linear correlation between hourly PM10 concentration and solar irradiance in our dataset was low (r ≈ -0.2), because many other factors (primarily cloud cover) dominate short-term irradiance variability. This low raw correlation might suggest PM10 as being unimportant, but that would be an oversimplification. In reality, even a relatively small increase in particulate matter can measurably reduce solar radiation reaching panels. Recent research in Korea demonstrated that a 10 µg/m³ increase in PM10 can cause about a 2.17 MWh reduction in hourly PV generation across the national PV fleet individual solar plants, this translates to noticeable power losses and revenue impacts. Our model captured this phenomenon: on days with very high PM10 (e.g., dusty or smoggy days), the forecast without PM10 input consistently over-predicted solar output, whereas the model with PM10 was able to adjust predictions downward, closer to reality. Thus, the value of including PM10 lies in better processing of those pollution events and subtle attenuation effects that a cloud-only model would miss. We have avoided attributing an exaggerated importance to PM10; rather, we incorporated it in a measured way. The neural network's learned weights indicated that on clear or moderately polluted days, PM10 exerted a minor influence, but on heavy pollution days, the PM10 feature weight increased in importance for the prediction. This aligns with physical expectations – aerosols matter most when their concentrations are high or during periods of low cloud cover when aerosol attenuation becomes the limiting factor for solar transmission. Additionally, PM10 can indirectly affect solar output via soiling (11) (dust deposition on panels), which accumulates over time. While our current model does not explicitly simulate soiling, some of the PM10 effect captured may include days-after impact (if the prior day's dust affects next day output). We addressed this by using lag features (e.g., previous day PM10) in the model, but a more explicit physics of soiling could be an avenue for future improvement.

5.3 MLP vs LSTM

In theory, an LSTM that can utilize temporal information should not do worse than an MLP that treats each hour independently. However, in our implementation, the LSTM was not given a true sequence to learn; effectively, it behaved like a complex MLP. The LSTM had roughly ~4,300 parameters (for 8 inputs and 64 hidden units plus gating mechanisms), compared to the MLP's ~ (8128 + 12864 + 64*1) ≈ 12,000 parameters. Hence the MLP actually had more trainable parameters, giving it a higher capacity to fit the data. Additionally, the MLP's direct connections might have been easier to train with the given data size, whereas the LSTM's gating could introduce optimization difficulty when not truly needed for a one-step input. In short, the LSTM was likely under-utilized. If we had instead fed – say - the past 24 hours of features to the LSTM for each prediction, it may have captured morning/afternoon transitions or day-to-day persistence better and potentially surpassed the accuracy of the MLP. Our result may hence serve as a cautionary tale: using an LSTM without appropriate sequence data can lead to worse performance than a simpler feed-forward model.

Another point to consider was that the solar power time series exhibited a very strong autocorrelation on 24-hour lags (because of the diurnal cycle and weather patterns). However, our random train-test split (or even time-based split) may not have fully challenged the model's ability to extrapolate temporal patterns beyond what the features already encoded. The features such as sun angles and irradiance basically informed the model the time of day and season, hence a lot of what an LSTM might learn (such as; "it tends to be sunny at noon") was already explicitly provided by the inputs. Therefore, the added value of sequence memory was low. The LSTM did not get to display its true potential in such a scenario. In fact, its slight underperformance might just be due to it having to learn to copy the input effectively (since at sequence length 1, an optimal LSTM would just transform inputs to output similar to how an MLP would, but it would not a direct one-to-one mapping – because the MLP also needed to unnecessarily manage an internal stage).

The MLP captured the needed mappings (from weather/PM to power) in a straightforward manner. The LSTM was a more complex tool applied to a problem that did not require the added complexity. This highlights an important point in ML implementation. The most sophisticated model is not always the most accurate, especially if misapplied. We suspect that had we structured the problem as a time-series forecasting (predict future hours from past hours without giving irradiance of the future as input), the LSTM or sequence models would then have outperformed the MLP. However, our setup was more of a 'nowcasting' (predict current output from current conditions) than that of a true forecast horizon beyond the present.

5.4 PGNN vs PINN

The fact that PGNN outperformed PINN suggested that *too much* physics guidance can be detrimental when the data has other complexities or when the physics model is imperfect. Our Beer–Lambert law is a simplification. It assumed a direct exponential relationship between irradiance and pollution for the impact on power. In reality, cloud cover, aerosol type, solar spectrum, and PV panel characteristics all affect the relationship. By forcing the network to adhere strongly to $I_0 e^{-\alpha C}$, the PINN might have led to underfitting some aspects of the data. Essentially, the PINN prioritized satisfying the Beer–Lambert equation, possibly at the cost of not minimizing the true error. This manifested as a consistent bias (underestimation at peaks, Figure 1), which increased the MSE.

The PGNN, with a lesser physics weight, was able to largely fit the data like the MLP did, only using physics as a slight regularizer to nudge it in plausible directions. It is telling that PGNN's error was only modestly higher than MLP's. This suggested that the inclusion of the physics term did not significantly decrease performance, which was a good sign — one can incorporate physics without severe penalty, as long as the weight is tuned properly. In fact, it could be interpreted that perhaps PGNN might generalize slightly better in unseen situations (e.g., if we had an extreme pollution event outside the training range, PGNN might have predicted a more physically reasonable drop in output whereas MLP might be more uncertain). Our test set, however, fell within the range of training distribution, hence the benefits of physics in the generalization were not significantly visible in the metrics.

The outcome reinforces the consensus in the literature that physics-informed models need careful calibration. If the physics is very accurate and the data is noisy, a higher weight assigned to the physics may return a higher accuracy. Conversely, if the physics is approximate and data is rich, a lower weight may be better. In our case, the Beer–Lambert law is approximate for PV output (because it doesn't include cloud effects explicitly, among other factors), hence a high weight may have caused greater mismatch. In practice, one could attempt to learn the optimal α or even learn a function for expected reduction rather than a fixed formula, to make the physics constraint more flexible.

Interestingly, one advantage that we noticed qualitatively was that the PINN's predictions never violated the physics intuition; i.e. it never predicted a higher output for higher PM (with the same irradiance) in training. The MLP, if left unconstrained, could in theory learn some spurious correlation where at very high PM it might still predict high output (especially if that high PM coincided with; say; clear winter days in the training data). We checked for such behavior: there was one instance in the training data with moderately high PM but also a very cold clear day, where output was high (cold temperatures can boost PV efficiency). The MLP fit that particular point well, whereas the PINN somewhat underpredicted it (because PM was high, it expected a drop). This nuance points out that not all deviations from the

Beer–Lambert law are "errors" – some are explained by other factors (such as temperature improving efficiency, or perhaps the PM measurement not fully correlating with the actual optical depth depending on particle size distribution, etc.). The PINN was not aligned with these explanations, hence it attributed any deviation to error, underpredicting when the reality was counteracting the pollution effect by another effect.

The PINN was behaving exactly as designed – it was prioritizing physical consistency. The higher error was a result of model bias – by imposing a bias (Beer–Lambert law), we constrained the solution space. If the true data-generating process had strictly followed that law, the PINN would excel. However, because the real system was additionally influenced by other factors (clouds, temperature, etc.), the PINN's bias led to a larger bias-variance tradeoff challenge. In essence, we traded some variance (overfitting) for bias (underfitting to one pattern), and in this case, bias had more effect on model accuracy than the variance did. However, the PINN's performance was not; by any means; catastrophic; an MSE of 35.6 vs 24.5 (MLP) on such a variable output is still reasonable. It simply was not the best model for predictive accuracy in this case.

5.5 Novelty and impact of using PM10 data

One of the key novelties of this work is demonstrating that including PM10 data improved solar power forecasting in Gwangju. This is in line with emerging research but still not widely adopted in operational models. Our results indicated that when PM10 was left out, the MLP's MSE increased by ~10%. The addition of PM10 was hence a significant improvement considering that it originated from a single additional feature. The implication is that in regions where air quality varies, solar forecasts can be materially improved by monitoring and correcting for pollution. For instance, on days with unexpected smog, a model aware of PM levels can reduce its prediction of PV output even if irradiance forecast from weather models is high, thus avoiding overestimation of solar generation for grid operators.

From a research perspective, our integration of PM data and physics is novel in the Korean context. Gwangju, like many cities, experiences Asian dust (yellow dust) events and local pollution. Our model provides a framework to quantify that effect. We demonstrated that the Beer–Lambert law can serve as a reasonable first-order physics model linking PM to solar attenuation. Prior work had shown correlation; we went a step further to actively use it in prediction.

It should be clarified that while a few studies globally have considered aerosol indices (like AOD) in irradiance models, it is still far from being implemented mainstream in solar forecasting, particularly using ground PM sensors and in a combined data-driven + physics manner. Our literature search did not find any studies that specifically built a PINN for solar output with Beer–Lambert law – this is, to our knowledge, a first. Thus, the methodological novelty lies in the physics-informed approach, and the application novelty lies in using local air pollution for PV prediction in our region.

5.6 Interpretation of model behavior

To further interpret model behavior, we examined feature importance and learned relationships. The MLP's learned weights (when transformed back to original scale) indicated that;

unsurprisingly; irradiance was the dominant feature, with a positive contribution to output. Sun altitude also had a positive effect (which correlated with irradiance as well). Cloud cover had a negative weight, meaning more clouds reduced output – consistent with physics. PM10 had a negative weight as well, which was relatively small compared to irradiance, but still non-negligible. For example, in the MLP's first layer, the weight on PM10 (standardized units) was ~ -0.15 (compared to +0.9 for irradiance's weight in that layer), showing that as PM increased, the network output tended to decrease, all else being equal. Hence, the MLP did learn the correct direction for the pollution effect from data alone. The PGNN and PINN, by design, enforced that relationship; indeed, their learned effective weight on PM10 was slightly stronger (for the PINN it was -0.2 in the first layer equivalent). This aligned with expectation: the PINN was more sensitive to PM changes because it was explicitly trained to be so.

We also noticed that the Irradiance² feature helped models capture a mild nonlinearity. Without it, the MLP's error was slightly higher and it tended to over-predict at very high irradiance (likely because PV output saturated due to temperature or inverter limits). With the squared term incorporated, the model could decrease its predictions at the top end of the range, better matching the actual flattening of power output on extremely sunny, hot afternoons.

5.7 Perspectives

Several improvements could be pursued in future studies.

5.7.1 *Better temporal modeling*

As noted, using an LSTM or sequence network appropriately could improve forecasting when predicting future time steps. One could feed past hours of data (including past outputs) to predict the next hour's output. This is more challenging because it then becomes a recursive forecasting model, needing to use predicted irradiance or weather forecast inputs. A variant of this is using sequence-to-sequence models or Temporal Convolutional Networks (TCN) that have shown promise in solar forecasting.

5.7.2 *Adaptive physics parameters*

Instead of a fixed α for Beer–Lambert, one could make α a function of sun altitude (the optical air mass) or even learn a small neural network to compute expected attenuation based on PM and maybe other factors (humidity can affect aerosol scattering). This could make the physics loss more accurate. For example, under high humidity, aerosols might swell and have more effect; a learned physics module could account for that.

5.7.3 *Additional physics constraints*

We could enforce that predicted power never exceeds what is physically possible given the solar irradiance times panel area and efficiency. We know the system's capacity (peak 866 kW observed), so we could cap predictions or include a penalty for exceeding those limits. Our models did not predict > 880 kW, but if a model were to demonstrate a large error, especially at the top of the range, incorporating that constraint would be logical.

5.7.4 *Inclusion of other pollutants*

PM10 was our focus, but PM2.5 or other pollutants (ozone, NO_2) might also indirectly correlate with haze. In our data, PM2.5 was not directly available, but since PM2.5 is often a fraction of PM10, including it if available could refine the attenuation model since fine particles vs coarse have different optical properties.

6. Conclusion

In this paper, we presented a comprehensive study on integrating air pollution data and physics-based constraints into solar energy forecasting for the city of Gwangju, South Korea. We built data-driven models (MLP, LSTM) as well as physics-informed models (PGNN, PINN) that incorporated the Beer–Lambert law of irradiance attenuation by particulate matter. Our results demonstrated that including PM10 concentrations as an input feature improved prediction accuracy, underscoring the importance of considering air quality in solar power forecasting. Perhaps counterintuitively, the simplest model (MLP) achieved the best performance on our test data, outperforming the more complex LSTM and the physics-informed models. This outcome emphasized that, in practice, a carefully tuned straightforward model can sometimes be more effective than sophisticated ones, especially when the latter are not optimally configured, or the data characteristics favor the simpler approach.

We also found that strong enforcement of physical laws (as in the PINN) led to larger errors if those laws are approximate or if they trade off against capturing other influencing factors. The PGNN with a moderate physics regularization struck a better balance, nearly matching the MLP's accuracy while still adhering to expected physical behavior to some extent. This finding contributes to the ongoing discourse in science-guided machine learning: it is important to calibrate the relative influence of theory and data. Pure learning can yield the best fit, but physics-informed learning yields interpretability and consistency – the challenge is in achieving both with minimal compromise.

From an application standpoint, our work is one of the first to quantitatively demonstrate the effect of particulate pollution on solar farm output in Korea using machine learning. The model can serve as a tool for grid operators in Gwangju to anticipate troughs in solar generation during bad air quality days, potentially allowing for proactive grid adjustments. Moreover, the methodology of blending physics and ML can be extended to other renewable energy domains (e.g., wind energy, where one might enforce energy conservation laws in a turbine's power curve prediction).

In conclusion, integrating domain knowledge such as the Beer–Lambert law into neural networks is a promising direction to enhance model reliability, but it has constraints. Our study provides a case example with detailed analysis, and the lessons learned can guide future efforts. We advocate for continued exploration of physics-guided machine learning in renewable energy forecasting – aiming for models that are not only accurate but also aligned with the underlying science.

Data repository

At Github. https://github.com/TheGitpilot/Solar-Energy-Forecasting-in-Gwangju-Using-PM10-Data-and-Physics-Informed-Neural-Networks

7. References

1. Tsai, W.-C., Tu, C.-S., Hong, C.-M., Lin, W.-M. (2023) *A review of state-of-the-art short-term forecasting models for solar-PV power generation.* Energies, 16 (14), 5436. https://doi.org/10.3390/en16145436

2. Wang, Y., Shi, M., Xie, P., Dong, Q., Jia, Y. Chen, L. et al. (2022) *Hybrid LSTM–SVR–BO model for short-term PV-power forecasting.* Energy Reports, 8, 3025-3035. https://doi.org/10.1016/j.egyr.2022.02.163

3. Song, Z., Wang, M. Yang, H. (2022) *Quantification of the impact of fine particulate matter on solar-energy resources and PV-system performance*. ACS Environmental Au, 2 (3), 275-286. https://doi.org/10.1021/acsenvironau.1c00048

4. Zhang, X., Zhang, M., Cui, Y. He, Y. (2022) *Estimation of daily ground-received global solar radiation using air-pollutant data*. Frontiers in Public Health, 10, 860107. https://doi.org/10.3389/fpubh.2022.860107

5. Mardani, M., Hoseinzadeh, S. Astiaso Garcia, D. (2024) *Developing particle-based models to predict solar-energy attenuation using long-term daily remote and local measurements*. Journal of Cleaner Production, 434, 139690. https://doi.org/10.1016/j.jclepro.2023.139690

6. Alfadda, A., Rahman, S., Pipattanasomporn, M. (2018) *Solar-irradiance forecast using aerosol measurements: A data-driven approach*. Solar Energy, 170, 924–939. https://doi.org/10.1016/j.solener.2018.05.089

7. Daw, A., Karpatne, A., Watkins, W., Read, J. Kumar, V. (2018) *Physics-guided Neural Networks (PGNN): An application in lake-temperature modelling*. arXiv 1710.11431. https://doi.org/10.48550/arXiv.1710.11431

8. Raissi, M., Perdikaris, P., Karniadakis, G.E. (2019) *Physics-informed neural networks: A deep-learning framework for solving forward and inverse problems involving nonlinear partial-differential equations*. Journal of Computational Physics, 378, 686-707. https://doi.org/10.1016/j.jcp.2018.10.045

9. Korea Meteorological Administration (2021) *Climate and air-pollution statistics of Gwangju*. Seoul: KMA. https://www.kma.go.kr

10. Korea Energy Economics Institute (2024) *Analysis of international new & renewable-energy policy and market*. Issue Paper 24-23. https://www.keei.re.kr

11. *SolarAnywhere® (2023) Particulate Matter In-depth: Support for soiling-loss modeling.* https://www.solaranywhere.com