Chapter 1

Introduction

1.1 INTRODUCTION

As the world searches for sustainable sources of energy, solar power has been at the forefront of the shift towards renewable energy sources. Yet the highly intermittent characteristics of solar radiation create a key barrier to efficient grid integration, scheduling of energy, and optimization of the system. One of the most important elements in solar power forecasting is Global Horizontal Irradiance (GHI), which is the total amount of per unit area shortwave radiation arriving at a horizontal surface. Global Horizontal Irradiance (GHI) is the total amount of solar radiation received per unit area on a horizontal surface at the Earth's surface. Precise GHI prediction is important for PV power generation forecasting, load dispatch planning, and energy reliability assurance.

Conventional GHI forecasting techniques usually depend on numerical weather prediction (NWP) models or sensor networks on the ground. Although useful in some situations, these methods are prone to issues like poor spatial resolution, delay, and a lack of ability to detect fast changes in atmospheric conditions such as cloud motion. In this project, we investigate the capability of deep learning structures, namely spatiotemporal autoencoders combined with ConvLSTM layers, to capture the time-evolution of cloud patterns through sequences of satellite images. Our methodology utilizes visible-channel data from the INSAT family of geostationary satellites, which offer regular and broad-area coverage of the Indian subcontinent.

By integrating convolutional neural networks (CNNs) for spatial feature learning with temporal dependency-capturing recurrent units such as ConvLSTMs, the approach learns to predict and reconstruct sequences of cloud-impaired images. From these reconstructed frames, cloud indices are calculated and transformed to matching GHI values using empirical equations.

The forecast of GHI is important to ensure the proper use of solar energy. GHI forecasting can provide better energy forecasts, enabling solar power plants to forecast energy production and supporting grid operators in balancing electricity supply and demand. It also provides grid stability by predicting changes in solar power due to cloudiness and atmospheric variations, minimizing instability. Moreover, precise GHI forecasts optimize the use of battery storage and minimize dependence on backup energy supplies, thus saving costs. In addition, predictive GHI information supports the improvement in the longevity of solar panels as operational modifications can be made to avoid overheating or underperforming.

1.2 IMPORTANCE OF SOLAR IRRADIANCE FORECASTING

The shift to renewable energy has brought along a new suite of challenges to power system planning and operation, foremost among them is power generation variability because of weather-dependent resources such as solar and wind. In contrast to traditional power plants that provide predictable and controllable output, solar photovoltaic (PV) systems are extremely sensitive to solar irradiance, which varies throughout the day depending on geography, time, season, and most importantly cloud cover. Global Horizontal Irradiance (GHI), or the sum of solar radiation received per unit area on a horizontal surface, is the most basic variable in solar energy forecasting. Reliable forecasting of GHI is essential for:

Power Grid Stability and Demand-Supply Balancing

Grid operators need to balance power production with real-time demand to provide frequency stability. A rapid decline in solar power generation due to cloud movement has the potential to destabilize the grid if not anticipated and accounted for beforehand. Short-term GHI forecasts assist in load forecasting, spinning reserve planning, and ramp rate control.

Renewable Energy Integration

As the proportion of renewables increases, especially in sun-abundant nations such as

India, balancing intermittent sources becomes increasingly challenging. Predicting GHI enables energy planners to plan backup generation (e.g., gas turbines, batteries) effectively, minimizing fossil fuel reliance and enhancing the reliability of renewable supply.

Optimizing PV System Operations

In rooftop PV systems and solar farms, forecasted real-time GHI improves Maximum Power Point Tracking (MPPT), control strategies for the inverter, and overall prediction of energy yields. Forecasted data can be utilized by operators to modify panel cleaning schedules and tilt angles and battery charging cycles.

1.3 GLOBAL HORIZONTAL IRRADIANCE

1.3.1 OVERVIEW

Global Horizontal Irradiance (GHI) is an important parameter for solar energy forecasting and is the total amount of direct and diffuse solar radiation that hits a horizontal surface on the Earth. It consists of two parts: Direct Normal Irradiance (DNI), or the sunshine arriving in a straight line from the sun, and Diffuse Horizontal Irradiance (DHI), or sunshine scattered by clouds, atmospheric molecules, and aerosols on its way to the surface. The mathematical expression for GHI is generally expressed as:

$$GHI = DHI + DNI \times \cos(\theta)$$

equation:i

where θ is the solar zenith angle, the position of the sun measured from the vertical.

GHI is arguably the most essential input to determine the output of solar photovoltaic (PV) systems, particularly those installed on flat-tilt roofs or flat rooftops that don't follow the sun's trajectory. GHI, therefore, directly influences how much energy a solar panel produces at a given moment. Therefore, GHI forecasting is a crucial component of operational energy management, particularly for grid-connected solar installations where

sudden irradiance variations on a short time scale can cause power mismatches, grid instability, and even energy curtailment.

Whereas surface-based sensors such as pyranometers offer instantaneous measurements of GHI, they are spatially limited and cannot be scaled up to large areas. Satellite imagery comes in handy here. Geostationary satellites like INSAT offer high-resolution, large-scale, and frequent visual data that captures the dynamic motion of clouds, one of the most powerful drivers of solar radiation variability. Through the analysis of satellite image sequences, one can deduce cloud cover and sky clarity, both of which have a direct influence on the diffuse and direct components of GHI.

1.3.2 CLOUD COVER ANALYSIS AND ITS EFFECT ON GHI PREDICTION

Cloud cover is among the most important factors influencing Global Horizontal Irradiance (GHI) forecasting since clouds have a direct impact on the quantity of solar radiation received at the Earth's surface. Thick cloud cover can absorb or scatter sunlight, decreasing direct normal irradiance (DNI) and enhancing diffuse horizontal irradiance (DHI), causing variations in total GHI. The effect of cloud cover depends on cloud type, height, density, and motion, which complicates precise forecasting.

1.4 PROBLEM STATEMENT

The precise prediction of Global Horizontal Irradiance (GHI) is important for the effective functioning of solar energy systems, but it is a very difficult task because of the unpredictable behavior of atmospheric conditions, mainly cloud formations and movements. Current ground-based techniques for GHI estimation are based on a sparse network of pyranometers and weather stations, which do not have the spatial density and real-time availability required for large-scale forecasting. While numerical weather prediction (NWP) models provide more comprehensive coverage, they tend to lack high resolution, are computationally expensive to run, and are less effective in reproducing

short-term, local cloud structures that are important in forecasting solar power generation, particularly in areas where weather patterns change quickly.

This project solves the particular problem of GHI estimation and forecasting through visible-band satellite images by building a spatiotemporal deep learning framework using autoencoders and ConvLSTM networks. The biggest challenge is to create a model that can learn from time sequences of satellite images to extract the history of cloud patterns over time and convert that history into precise irradiance predictions.

1.5 OBJECTIVES

The main goal of this project is to create a precise and effective deep learning-based framework for the prediction of Global Horizontal Irradiance (GHI) from visible-band satellite imagery. The framework will seek to leverage spatiotemporal patterns in sequences of satellite images to predict solar irradiance over a given area, independent of the need for dense ground-based sensor networks. This method is designed to overcome the shortcomings of traditional GHI forecasting techniques through the use of state-of-the-art computer vision and sequence modeling algorithms.

The main goal is to develop and deploy a spatiotemporal autoencoder architecture that can learn useful representations from high-resolution satellite imagery. The autoencoder should be able to compress and reconstruct input frames and preserve important cloud and atmospheric information that affects solar irradiance. To model temporal cloud evolution, the model incorporates Convolutional Long Short-Term Memory (ConvLSTM) layers, which can capture dynamic change between successive image frames and improve the predictive strength of the network.

1.6 SCOPE OF THE PROJECT

The scope of this project includes the design, development, and testing of a deep learning-based framework for Global Horizontal Irradiance (GHI) prediction from visible satellite imagery. The model trained will be efficient enough to produce forecasts with low latency and hence suitable for applications in solar energy forecasting, grid load balancing, smart grid management, and climate monitoring systems The project aims at utilizing high-resolution satellite images from INSAT to derive cloud-related features that affect solar radiation, and utilize these features to estimate GHI over a specified geographic area.

The technical scope contains preprocessing and collection of satellite data, especially from HDF5 files, and then moving to the spatiotemporal neural network architecture design using autoencoders for learning spatial features and ConvLSTM layers for temporal sequence modeling. The system will be trained and validated over a structured dataset containing image sequences and relevant ground-truth values of GHI. The system also incorporates a cloud index calculation module, which is employed to indirectly calculate irradiance levels from visual patterns in cloud cover.

Chapter 2

Background

2.1 LITERATURE REVIEW

Validation of global irradiance derived from INSAT-3D over India

The article examines the accuracy of global horizontal irradiance (GHI) obtained from the INSAT-3D satellite over India and compares it with pyranometer measurements on the ground. Approach includes deriving GHI values through radiative transfer models that are implemented on INSAT-3D satellite data. The obtained values are cross-validated by ground observations at various sites spread over different climatic zones of India. Statistical parameters like mean bias error (MBE), root mean square error (RMSE), and correlation coefficients are employed to check for accuracy and reliability of the satellite-derived data.

The outcomes show that INSAT-3D gives a fair approximation of GHI with different levels of accuracy based on cloud cover and atmospheric conditions. Although the satellite-based method yields high spatial coverage, retrieval algorithms need to be improved to maximize accuracy. The results validate the suitability of INSAT-3D data for solar energy uses while pointing to the necessity of ongoing validation and improvement.[15]

Short-term Solar Power Forecasting Using Satellite Images

This work suggests a model for forecasting using satellite images to predict solar power output at 15-minute steps by estimating cloud index values. The technique uses processing cloud index images from ISRO-SAC satellite data and extracting global horizontal irradiance (GHI) values. Cloud motion is estimated using an enhanced block search algorithm and utilized to predict irradiance. A Long Short-Term Memory (LSTM) model is trained with historical GHI values to predict power output in the short term. The findings show that the LSTM-based method greatly enhances short-term solar power forecasting accuracy over traditional models. [11]

Intercomparison of Satellite-Derived Solar Irradiance from the GEO-KOMSAT-2A and HIMAWARI-8/9 Satellites by the Evaluation with Ground Observations

The study compares the performance of the GK-2A satellite-optimized UASIBS-KIER model with the HIMAWARI-8/9 satellite-derived solar irradiance models. The approach is to modify the UASIBS-KIER model to include GK-2A satellite data and compare its results with ground pyranometer measurements. The NMSC-INS and JAXA-INS models are also included for comparison. Cloud detection and classification methods, and radiative transfer models, are used to estimate solar irradiance. The performance of the model is evaluated using statistical indicators like Mean Bias Error (MBE) and Root Mean Square Error (RMSE) at instantaneous, hourly, and daily timescales.

The results showed that the new UASIBS-KIER model yields higher accuracy compared to earlier models, particularly during clear-sky conditions. Although the work verifies the use of satellite-derived irradiance in solar energy applications, additional improvements are needed to improve reliability in operational forecasting.[5]

Solar Radiation Estimations Over India Using Meteosat Satellite Images

Precise solar radiation estimation is crucial for estimating solar energy potential in India, where there exists high solar irradiance but modified by atmospheric aerosols. The research analyzes the temporal and spatial differences of solar radiation while emphasizing the significance of aerosols in the evaluation of solar resources. The approach consists of obtaining global horizontal irradiance (GHI) and direct normal irradiance (DNI) from the IrSOLaV/CIEMAT satellite model, which combines clear-sky models, cloud index analysis, and MODIS satellite data aerosol optical depth (AOD) corrections. The calculated values are tested against ground observation from the World Radiation Data Centre (WRDC) database, taking into account biases and uncertainty sources under various climatic conditions.

The outcomes show that the model slightly overestimates GHI in relation to ground

observations, with 5% bias and about 12% RMSE. Discrepancies are larger in Trivandrum owing to local uncertainties of measurement. The research stresses the need for reliable aerosol and water vapor data in order to enhance the accuracy of estimation. Though the model adequately accounts for seasonal variations in solar radiation, improved cloud detection and aerosol correction schemes are necessary to achieve greater accuracy in solar resource estimation.[14]

Distributed Solar Generation Forecasting Using Attention-Based Deep Neural Networks for Cloud Movement Prediction

The approach includes training a convolutional Long Short-Term Memory (ConvLSTM) network with attention mechanisms, such as a Convolutional Block Attention Module (CBAM) and a Self-Attention (SA) mechanism. The cloud forecast is subsequently fed as inputs into solar power forecasting models, such as Convolutional Neural Networks (CNN), and LSTM networks, to produce power output predictions. The models are tested against baseline ConvLSTM and persistence-based models. Experiment results show that attention-based ConvLSTM models substantially enhance the accuracy of forecasting cloud movement, especially for clouds at high altitudes, thus resulting in more accurate short-term solar power prediction. CBAM ConvLSTM shows maximum improvement, and forecast skill scores are improved by a maximum of 5.86% compared to non-attention models.[3]

Estimating Multidirectional Cloud Movements from a Single Sky Camera Using Directional Statistics

The method consists of taking pictures of the sky every five seconds with a Raspberry Pi Camera Module. Optical flow methods such as Lucas-Kanade, Farneback, TV-L1, and DeepFlow are compared to ascertain the most appropriate method for cloud movement estimation. The research provides Circular Standard Deviation (CSD) as a statistical measure to distinguish between one-directional and multi-directional cloud movements. Threshold-based is applied to categorize cloud motion patterns over time, with DeepFlow chosen as the best method. Experiments demonstrate that DeepFlow, in conjunction with

CSD, is able to detect one-to-two directional cloud motion transitions with great accuracy. Cloud layer separation and motion estimation are emphasized as key factors in solar energy applications by the research. The drawbacks, however, involve difficulties in the detection of more detailed cloud structures and possible errors in highly changing cloud conditions. The method is a cost-efficient alternative to multi-camera systems, and it offers a promising real-time cloud movement estimation solution.[7]

A Cloud Motion Estimation Method Based on Cloud Image Depth Feature Matching

This paper introduces a new cloud motion vector (CMV) estimation approach that utilizes depth feature matching to enhance accuracy and robustness in dealing with complicated cloud situations. The method uses a multi-step process that comprises image improvement, self-supervised feature extraction, matching of features, fusion of features, and spatiotemporal filtering. Depth features are derived from temporal cloud images without motion-labeled training data, thus the approach is independent of supervised learning. The features extracted are matched over time-series images to estimate CMVs, and filtering is performed to improve motion vectors, minimizing noise and inconsistency.

Results show that the suggested method substantially outperforms conventional CMV estimation methods, especially in situations with intricate cloud structures. The method enhances precision and stability, providing accurate motion estimation in varied atmospheric conditions. Limitations exist in possible computational overhead and a requirement for further optimization under severe weather conditions. The research provides an innovative CMV estimation system that enhances solar radiation forecasting and meteorological purposes.[1]

Forecasting Photovoltaic Power Generation Using Satellite Images

In this, approach is two-step: one, a cloud amount prediction network using Eidetic 3D Long Short-Term Memory (E3D-LSTM) processes historical satellite images to predict short-term cloud cover. Two, the predicted cloud amount is fed into a convolutional self-attention-based LSTM model that incorporates historical weather, solar elevation angles,

and cloud predictions to produce PV power forecasts. The performance of this model is benchmarked against classical forecasting models which make no use of cloud predictions. Results indicate that the inclusion of cloud amount forecasts decreases the Mean Absolute Percentage Error (MAPE) by 22.5% in comparison to models without cloud forecasting. The model is good under changing weather conditions but has limitations in extreme cases with high atmospheric changes. Although the proposed method greatly enhances PV power forecasting accuracy, more improvements in cloud classification and prediction algorithms are required for real-time grid applications.[4]

Prediction of Solar Power Using Near-Real-Time Satellite Data

The approach is based on processing near-real-time visible images from the Himawari-8 satellite to estimate cloud motion vectors via optical flow methods. The vectors are utilized for tracking cloud advection and forecasting future cloud cover. The model translates satellite reflectance data into cloud index values, which are then projected to the clear-sky index using historical fitting functions. Lastly, the forecasted global horizontal irradiance (GHI) is then translated into power estimates based on site-specific power conversion models that are trained with past data. The model was tested with real-time data from four solar farms in Australia. Results show that the new approach performs better than the persistence model, with errors decreased by over 50% in short-term forecasts.

The model obtains less than 10% error in 34–60% of predictions, though accuracy suffers under rapidly evolving cloud conditions. Main limitations are increased forecast errors resulting from satellite image latency and complexity in cloud index to irradiance conversion. While the model enhances solar power prediction accuracy, improvement in cloud detection and irradiance conversion algorithms is required for real-time operational implementation.[8]

Satellite-Based Operational Solar Irradiance Forecast for Uruguay's Solar Power Plants

The approach relies upon the Cloud Motion Vector (CMV) satellite method, predicting

cloud motion at various spatial resolutions. The model includes single-pixel and space-averaged CMV forecasts that are subsequently merged with space-averaged persistence in order to produce more detailed final irradiance forecasts. The forecast is issued at 10-minute intervals. Results show the suggested method enhancing short-term forecasting of solar power over persistence models, supporting grid operators in terms of power dispatch and reserve scheduling. Limitations, however, are difficulties in precise estimation of cloud motion, errors caused by high temporal and spatial variability, and reliance on satellite image quality. In spite of these drawbacks, the paper points to CMV-based satellite forecasting as enhancing solar power integration into the grid.[9]

.

Solar Power Prediction Based on Satellite Images and Support Vector Machine

This work proposes a model for predicting solar power using satellite imagery and a support vector machine (SVM) learning approach to enhance precision in forecasting. The model exploits atmospheric motion vectors (AMVs) derived from satellite imagery to monitor cloud motions and predict fluctuations in solar irradiance.

The approach is to analyze four years of historical satellite imagery to create a large dataset for the training of SVM. The model trained is then employed to forecast solar power generation based on learning the past patterns from satellite images and their respective power generation. The performance of the SVM model is compared with traditional time-series models and artificial neural network (ANN) models to assess prediction accuracy.

The findings show that the suggested method performs better than conventional models for short-term solar power forecasting. In spite of the advances in prediction precision, the model is confronted with issues like high temporal and spatial variability, errors in cloud motion estimation, and satellite image quality dependency. These issues impact the credibility of forecasts in fast-changing weather patterns. However, the research proves the efficiency of using satellite images coupled with machine learning methods to improve solar power forecasting.[12]

Cloud Cover Forecast Based on Correlation Analysis on Satellite Images for Short-Term Photovoltaic Power Forecasting

The approach used is processing visible and infrared satellite images and transforming them into numerical data and feeding them into an artificial neural network (ANN) model for power forecasting. The ANN model, which has two hidden layers with 10 neurons each, was trained using a variety of input variable combinations — cloud cover, visible images, infrared images, and all three.

The findings show that the best forecasting performance throughout all test periods was obtained by the multivariable model that included all three data sources. Out of single data sources, cloud cover data resulted in consistent performance for short-term forecasts, whereas visible images were best for ultra-short-term forecasting. The research, however, points out constraints such as possible error in the extracted cloud data and less accuracy during fast-changing weather situations. In spite of these difficulties, the suggested method presents a good solution for improving the reliability of PV power forecasts based on satellite data.[2]

LSTM–GAN based cloud movement prediction in satellite images for PV forecast

The paper proposes a hybrid deep learning method combining Long Short-Term Memory (LSTM) and Generative Adversarial Networks (GAN) for predicting cloud movement in satellite images to enhance photovoltaic (PV) power forecasting. The GAN component generates future cloud images using random latent vectors, while the LSTM model identifies temporal patterns in sequential satellite images. The methodology leverages a dataset of 30,507 infrared satellite images captured every 15 minutes by Korea's Communication, Ocean, and Meteorological Satellite 1. The model was compared with alternative hybrid models such as CNN-ANN, CNN-LSTM, GRU-GAN, and BILSTM-GAN.

The results indicate that the LSTM-GAN model achieved superior prediction accuracy compared to the other tested models. However, the study acknowledges limitations, including the model's dependency on high-quality image data and potential challenges in handling extreme weather variations. Despite these constraints, the proposed model demonstrates improved forecasting capabilities, supporting more stable PV power generation predictions.[6]

2.2 THEORETICAL BACKGROUND

The Indian National Satellite System (INSAT) series is a group of geostationary meteorological satellites owned by the Indian Space Research Organisation (ISRO). Satellites such as INSAT-3D and INSAT-3DR offer vital weather-related information, such as multi-channel images in visible, infrared (IR), and water vapor bands. These satellites record images at 15- 30 minutes' temporal resolution and up to 1 km spatial resolution, making them appropriate for near real-time monitoring of atmospheric phenomena.

INSAT satellites produce data in the form of HDF5, which contains more than one layer of spectral and derived data. Visible imagery is notably significant during daytime for monitoring cloud cover, motion, and intensity—critical parameters directly affecting solar irradiance. The figure given below is a sample clipped image obtained from INSAT 3D database. The dataset is 2019 cloud image in every 30min time frame that has various attributes like IMG_VIS, brightness, cloud temperature etc. we use image visibility for the GHI prediction throughout the project.

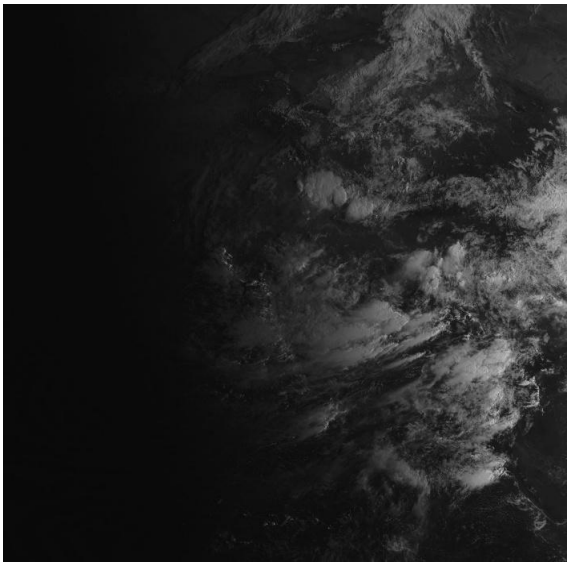

Figure 2.1: Raw Data

2.3 TECHNICAL BACKGROUND

2.3.1 AUTOENCODERS AND TEMPORAL SEQUENCE LEARNING

Autoencoders are a type of unsupervised deep learning that is meant to learn compact, compressed representations of input data—commonly known as latent embedding or feature codes. An autoencoder is made up of two main parts: the encoder that maps the input into a lower dimension, and the decoder that decodes the original data from this latent representation. In satellite image processing, autoencoders work very well to learn meaningful spatial patterns like cloud structures, textures, and features related to irradiance without labeled data.

2.3.2 INTRODUCTION TO CONVOLUTIONAL LSTM (CONVLSTM)

Convolutional LSTM (ConvLSTM) is a deep learning architecture that is custom-designed to bring together the spatial learning strengths of Convolutional Neural Networks (CNNs) and the temporal modeling abilities of Long Short-Term Memory (LSTM) networks.

The main concept of ConvLSTM is to substitute the fully connected matrix multiplications in a regular LSTM cell with convolutional computations. This enables the model to keep the spatial information of the data during the temporal modeling process. Rather than flattening an image into a vector, ConvLSTM takes input 3D tensors with size matching height, width, and channels. This architecture is specifically most appropriate for satellite-based applications like cloud motion forecasting, and solar irradiance estimation where both the organization of pixels (spatial information) and how they change over time (temporal information) are important.

A second strength of ConvLSTM is parameter efficiency. Owing to shared spatial convolution operations, the quantity of learnable parameters is highly minimized relative to fully connected LSTM layers, leading to ConvLSTM models with greater scalability and reduced susceptibility to overfitting. Additionally, they can better generalize between varied regions or times because they have the capability of concentrating on neighboring motion patterns like drifting clouds or changing weather fronts.

ConvLSTM can be used in several layers, stacked hierarchically to extract low-level motion patterns in early layers and high-level spatiotemporal abstractions in deeper ones. When incorporated into autoencoder structures, ConvLSTM is usually located in the bottleneck layer to compress the temporal dynamics of input sequences, thus facilitating future frame prediction or direct regression to physical variables such as GHI.

2.3.3 SPATIO TEMPORAL FEATURE EXTRACTION CONCEPT

Spatiotemporal feature extraction is key in modeling solar irradiance from satellite images by capturing both spatial and temporal patterns. Spatial features—like cloud shape, size, and density—are extracted using CNNs from individual frames. Temporal features—such as cloud movement and evolution—are captured using models like RNNs, LSTMs, or ConvLSTMs. By combining these, models can learn both appearance and motion, enabling accurate forecasting. Architectures like ConvLSTM, 3D CNNs, and transformers help capture these inter-frame dependencies for improved prediction.

2.4 COMPARATIVE STUDY OF CLUSTERING VS AUTOENCODER

Global Horizontal Irradiance (GHI) forecasting is a cornerstone for efficient solar energy management and grid stability. Accurate estimation of GHI requires understanding and quantifying the influence of clouds on solar radiation. One of the primary methods used to estimate cloud influence is through the cloud index, a parameter derived by comparing actual cloud cover to clear-sky conditions. Initially, we adopted a clustering-based approach to derive this index from satellite images, attempting to classify and quantify cloud coverage over time. However, this method had several critical limitations that impeded its accuracy and temporal consistency. To overcome these challenges, we transitioned to a spatiotemporal modeling strategy using a ConvLSTM-based autoencoder framework. This document outlines the methodology, drawbacks, and performance of the clustering approach, and contrasts it with the significantly improved outcomes of the ConvLSTM-based model.Figures given below from 2.2 to 2.4 are cloud clustered images of the insat dataset along with their accuracy scores in table 2.1.

Clustering-Based Cloud Index Estimation

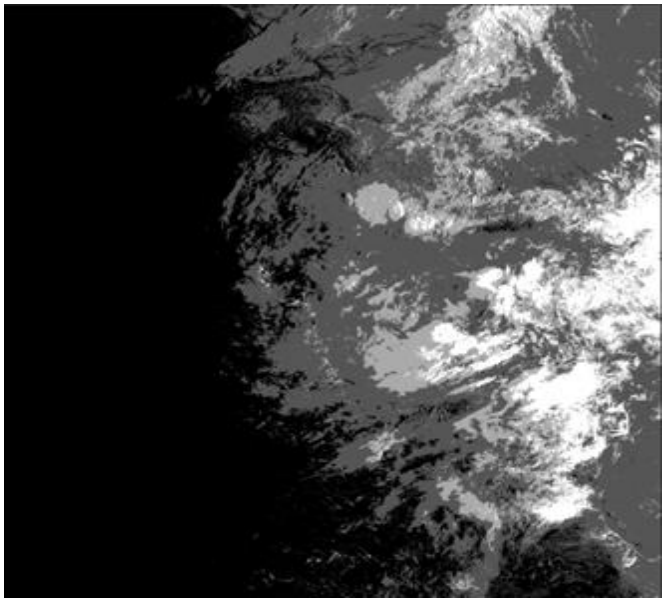

Figure 2.2: K Means Clustered Cloud Image

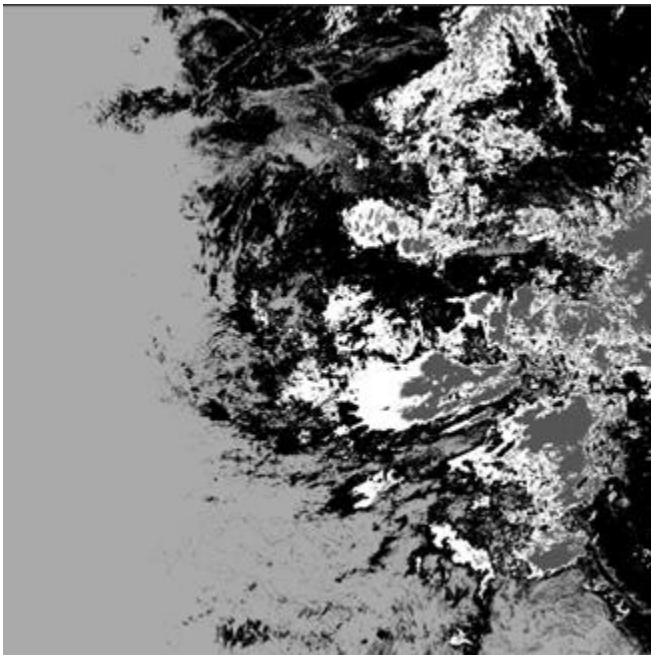

Figure 2.3: Fuzzy K Means Clustered Cloud Image

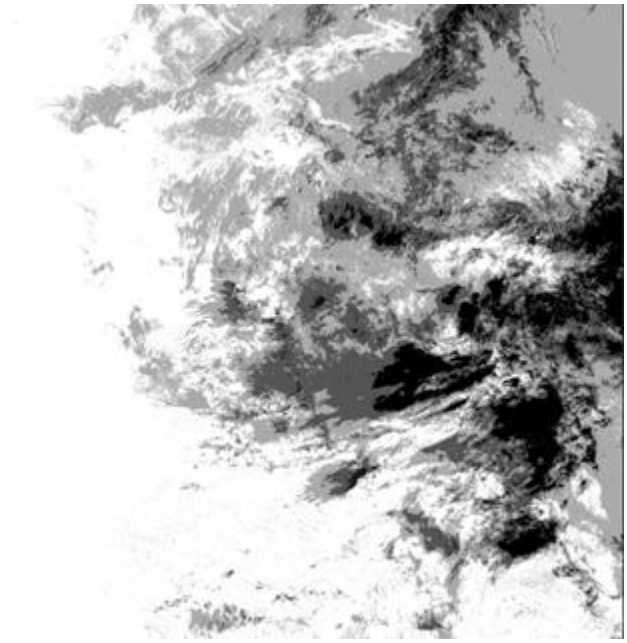

Figure 2.4: Gaussian Mixture Model Clustered Cloud Image

Clustering Method	Silhouette Score	Davies-Bouldin Index	Calinski-Harabasz Index
Fuzzy K-Means	0.6687	0.5095	459606.1878
K-Means	0.6565	0.5135	113171.5131
Gaussian Mixture Model (GMM)	0.6187	3.9011	20002.0257

Table 2.1: Clustering Scores

The table given in 2.1 is the evaluation matrix of clustering of cloud images. From the above table we can conclude that Fuzzy K-Means model gives best accuracy but as the clipping of Area of Interest cannot be calculated from INSAT Data due to lack of critical details of smaller precise region pixel clarity, this method is not reliable. The cloud index calculated using this method is 0.00 which is incorrect and cannot be further used to predict GHI. The equation used to calculate cloud index is given in equation ii.

$$\boldsymbol{Cloud\ Index} = \mathbf{1} - \frac{\boldsymbol{N} - \boldsymbol{N_{min}}}{\boldsymbol{N_{max}} - \boldsymbol{N_{min}}}$$

equation: ii

where 'n' represents the mean cloud pixel intensity in the image. This approach involved clipping the area of interest (AOI) and computing n_min and n_max across the cloudy classes.

Transition to Spatiotemporal Autoencoder-Based ConvLSTM Model

To address these shortcomings, we developed a spatiotemporal deep learning model based on Convolutional Long Short-Term Memory (ConvLSTM) layers, embedded within an autoencoder architecture. Unlike clustering, this model processes sequences of satellite cloud masks while preserving the spatial and temporal dynamics of cloud movement. The architecture takes in a sequence of six past satellite images, each spaced thirty minutes apart, and predicts four future GHI values, covering a two-hour prediction horizon. Each input sequence includes not only cloud mask data but also the corresponding clear-sky GHI and temporal encodings derived from sine and cosine transformations of time features.

The encoder-decoder structure of the model compresses the spatiotemporal information into a latent representation, allowing it to reconstruct and forecast future GHI values effectively. The key advantage of this approach is its ability to implicitly learn the relationship between cloud cover and solar irradiance, thereby eliminating the need for manually defined cloud index thresholds. The model outputs smooth and temporally consistent GHI values, which better align with real-world measurements.

Chapter 3

METHODOLOGY

3.1 DATA PRE PROCESSING

- The raw data is kept in HDF5 (Hierarchical Data Format version 5) files, which are very convenient to keep big and complex data, such as multi-dimensional satellite imagery.
- Programmatically manipulating the HDF5 structure to retrieve only the visible spectrum images that are directly related to cloud movement and the level of solar radiation. Separating the desired channels, and combining them into a sequential time-stamped structure is to maintain temporal continuity.
- After extraction, a region clipping procedure is hence applied to clip images spatially and keep only the geographic area applicable to the task of GHI prediction. This clipping operation not only minimizes computational complexity but also makes the model learn localized atmospheric patterns more accurately.
- After clipping Tirupathi region, the pixel intensity values of the images tend to vary with respect to sunlight intensity, atmospheric scattering, and seasonality changes. To compensate for these variations and maintain consistency across the dataset, image normalization is performed. This generally constitutes scaling the pixel values to an invariant range like 0 through 1 with min-max normalization or standardization with z-score normalization.

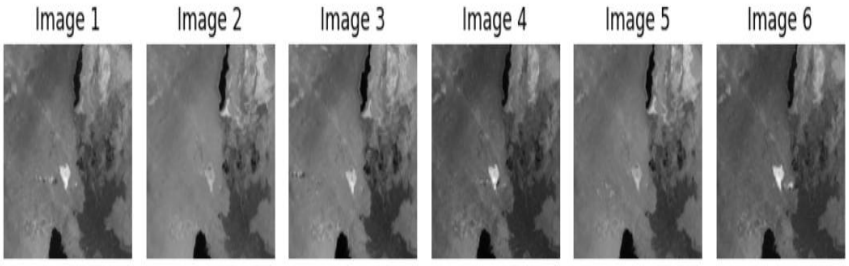

Figure 3.1: Stacked images after clipping

The image given in 3.1 is the final clipped images that are taken into one spatio-temporal cube and is used in the further prediction pipeline.

3.2 CHALLENGES IN DATA PREPROCESSING

- Dealing with hundreds of thousands of hdf5 files over extended periods of months or years of hourly satellite acquisition grows more complicated. Without an organized directory hierarchy, systematic naming conventions, and tagging of metadata, retrieval of particular sequences or frames for training models becomes inefficient.
- Preserving chronological order, mislabeling or disorganization in file hierarchy could distort the entire temporal timeline.
- Data pipeline includes several preprocessing operations like cropping, normalization, and resizing frames, each one of which creates new versions of the dataset that need to be properly connected back to their source.
- If there are missing sequential frames in a time series, the model fails to learn the temporal dependencies and motion patterns of clouds, which results in poor predictions. Preprocessing pipelines should thus incorporate solid methods to detect and fill these gaps in the data.

3.3 SYSTEM ARCHITECTURE

3.3.1 SPATIO TEMPORAL AUTOENCODER DESIGN

The Spatiotemporal Autoencoder design is the crux of the proposed Global Horizontal Irradiance (GHI) prediction approach from satellite images. This model is specifically designed to capture both spatial patterns (cloud shapes and distribution within an individual image) and temporal patterns (cloud movements over time). In order to do so, the model takes input data in the form of spatiotemporal cubes and essentially utilizes the strengths of both Convolutional Neural Networks (CNNs) and Convolutional Long Short-Term Memory (ConvLSTM) networks.

Input Representation: Spatiotemporal Cubes

The input to the autoencoder is not an individual satellite image but a sequence of images recorded at constant time intervals (30 minutes). The images are combined on top of one another to create a 4D tensor or a spatiotemporal cube of dimensions (T, H, W, C) where T is the number of time steps (6 frames), H and W are the height and width of an image (256×256 pixels), C is the number of channels.

This cube allows a complete perspective of cloud dynamics through time so that the model can pick up on how cloud patterns change in space and time.

Encoder: Hierarchical Feature Extraction

The encoder consists of several convolutional layers that are specifically aimed at extracting progressively abstract and higher-order spatial features. As the data progresses through these layers:

- The spatial resolution is gradually decreased through pooling or stridden convolutions.
- The depth grows, allowing the network to learn sophisticated features such as cloud density, structure, and movement vectors.
- The encoder does not only flatten the data but preserves the temporal dimension, meaning that all frames in the sequence are processed in a uniform manner.

Temporal Encoding through ConvLSTM

The feature maps encoded at all time steps are fed into a ConvLSTM layer, the temporal memory unit of the model. The ConvLSTM:

- Processes spatial feature maps per time step,
- Holds spatial coherence while capturing temporal relationships,
- Learns cloud development, motion patterns, and atmospheric changes,
- Saves temporal data in cell and hidden states, which change over time depending on input sequences.

In contrast to regular LSTM units which collapse data into 1D vectors, ConvLSTM operates on the whole spatial grids. This helps maintain local spatial patterns like a patch

of cloud moving towards the east in a correct and interpretable form in a temporal context.

Decoder: Frame Reconstruction

The decoder's goal is to generate the subsequent frame of the satellite image sequence, which is the immediate future. It contains:

- Upsampling layers
- Transposed convolutional layers.
- Connections from matching encoder layers to restore fine-grained spatial detail that could be lost during downsampling.

These operations incrementally restore the spatial resolution and decrease the depth, ultimately producing a frame of equivalent size and structure to the original inputs.

Predicted Output and GHI Relevance

The output frame is a forecast future satellite image. From the forecast frame, cloud indices or transmittance parameters can be calculated, which correspond directly to GHI estimation. As irradiance is greatly influenced by the presence and thickness of clouds, proper forecasting of cloud cover allows effective forecasting of solar energy availability. The benefit of this architecture is that the model does not depend exclusively on current or single frames but learns transitions, patterns, and temporal dynamics, resulting in improved generalization and accuracy particularly over various regions or weather conditions.

3.3.2 INPUT AND OUTPUT DIMENSIONS OF THE MODEL

Stage	Description	Tensor shape
Input	Spatiotemporal cube	(6, 256, 256, 1)
Encoder Output	Compressed features	(6, 64, 64, 128)
ConvLSTM Output	Temporal representation	(64, 64, 128)
Decoder Output	Predicted next satellite frame	(256, 256, 1)

Table 3.1: Input and Output Dimensions of the Model

3.3.3 ARCHITECTURE DIAGRAM

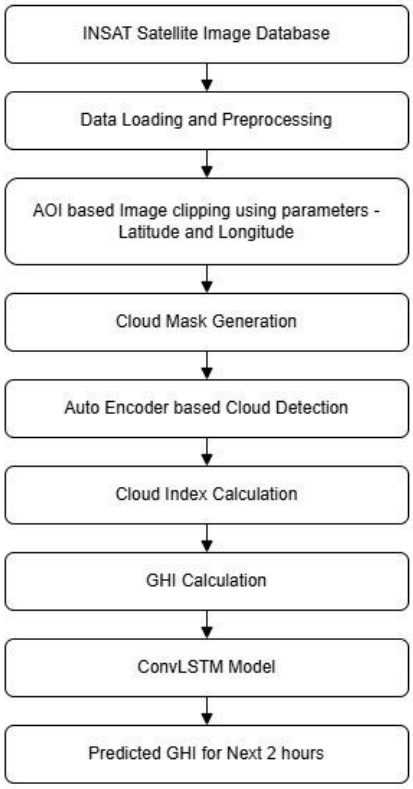

Figure 3.2: Architecture Diagram

3.3.4 CONVLSTM INTEGRATION STRATEGY

In the case of Global Horizontal Irradiance (GHI) forecasting from satellite images, ConvLSTM (Convolutional Long Short-Term Memory) is pivotal in capturing the temporal progression of spatial patterns notably the motion and change of clouds over time. The integration strategy of ConvLSTM in the spatiotemporal autoencoder architecture is carefully crafted to maintain spatial integrity while capturing temporal dependencies, providing an effective tool for short-term solar irradiance forecasting.

The satellite images employed in this project contain dense spatial information regarding cloud structures, but to predict GHI, one needs to know how these clouds evolve and shift over time. This is where ConvLSTM becomes crucial — it learns temporal patterns in the

spatial features learned from the satellite sequences.

In contrast to conventional LSTM layers that flatten spatial information, ConvLSTM layers preserve the spatial structure by performing convolution operations inside the recurrent gates. This allows the network to identify and learn how certain spatial features change over time.

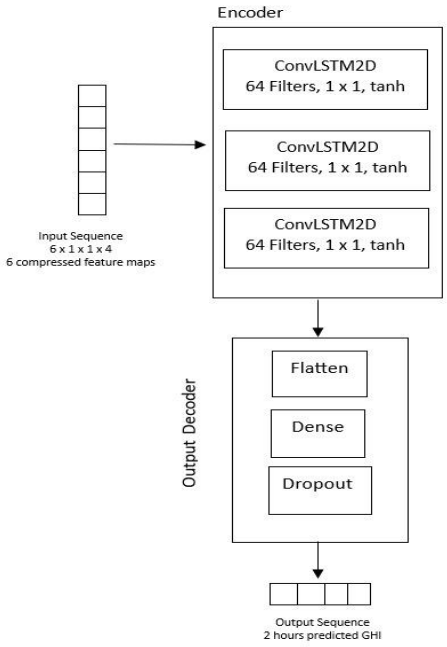

Figure 3.3: ConvLSTM Architecture

Technical steps involved in integration of ConvLSTM

- Input to ConvLSTM: The encoder gives a 4D tensor for every frame in the sequence (shape (batch_size, 10, 64, 64, 128) where 10 is the number of time steps). This sequence of tensors is input into a ConvLSTM layer for the learning of spatial-temporal dependencies.
- ConvLSTM Configuration:
    - Kernel size: Specifies the spatial extent for every convolution operation (e.g., 3×3).
    - Number of filters: Regulates the dimensionality of the output space (e.g., 64, 128).

- Activation: Usually tanh and sigmoid for gates.
- Return sequences: On or off depending on whether intermediate outputs are needed.

➢ Output of ConvLSTM: Creates a single or sequence of feature maps with embedded temporal knowledge. This temporally-enhanced representation is subsequently fed to the decoder.

➢ Decoder Processing: Uses upsampling and transposed convolutions to construct the estimated next frame from ConvLSTM output information.

3.4 TEMPORAL SEQUENCE MODELING WITH CONVLSTM

Spatiotemporal feature extraction is key in modeling solar irradiance from satellite images by capturing both spatial and temporal patterns. Spatial features—like cloud shape, size, and density—are extracted using CNNs from individual frames. Temporal features—such as cloud movement and evolution—are captured using models like RNNs, LSTMs, or ConvLSTMs. By combining these, models can learn both appearance and motion, enabling accurate forecasting. Architectures like ConvLSTM, 3D CNNs, and transformers help capture these inter-frame dependencies for improved prediction.

3.5 CLOUD PROCESSING AND GHI APPROXIMATION PIPELINE

One of the most critical elements in the GHI prediction algorithm is the cloud processing from satellite images to estimate how much impact clouds have on solar irradiance levels. Clouds and their motion are the foremost drivers of GHI variability, so processing clouds is an essential step in the whole prediction chain. To quantify the relationship, the system employs a proxy variable of the Cloud Index (CI) mapping visual cloud characteristics to their effects on irradiance values. The cloud index is usually calculated from satellite reflectance values derived from visible spectrum images, since higher reflectance suggests denser or thicker cloud cover. This index is a normalized quantity from clear sky (low reflectance) to fully overcast (high reflectance), which enables the model to correlate pixel-level cloud information with corresponding decreases in solar irradiance.

3.5.1 CLOUD MASK GENERATION USING AUTOENCODER

The initial process in the GHI prediction pipeline is the detection of cloud-covered areas in satellite images. This is done through the utilization of the reconstruction power of an autoencoder, which is learned to recognize patterns of cloud absence and presence. The result of this process is a cloud mask, a binary or probabilistic image that represents the probability of cloud coverage at every pixel.

Autoencoder is trained to learn a compact, efficient representation of input data and then reconstruct the input as closely as possible. When trained from a diversified dataset with heterogeneous cloud coverage, the autoencoder learns to encode and generate the common characteristics of the atmosphere, including clear sky, partial cloud cover, and dense clouds. Figure 3.4 is the cloud mask obtained after preprocessing.

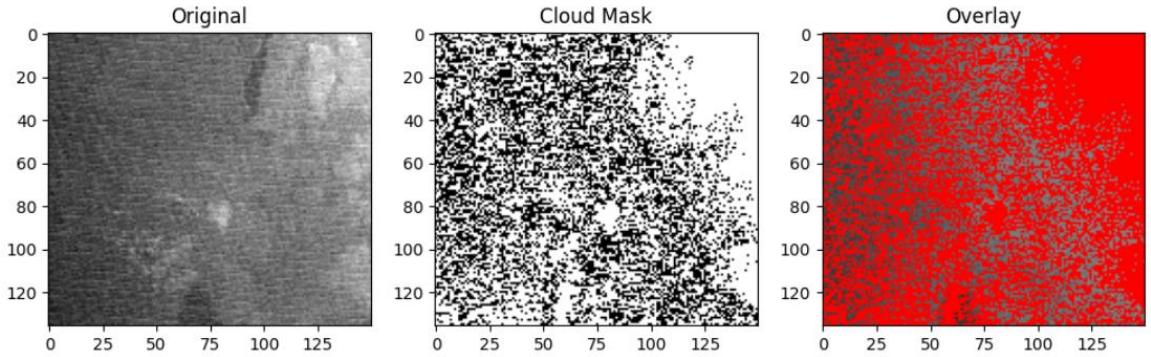

Figure 3.4: Cloud mask and overlay

3.5.2 CLOUD INDEX COMPUTATION

Once a cloud mask is produced by employing the autoencoder reconstruction error, the subsequent key step involves the quantification of the influence of clouds on solar radiation. This is achieved through the calculation of a Cloud Index (CI), which serves as a scalar adjustment for theoretical clear-sky irradiance and aids in the approximation of actual GHI (Global Horizontal Irradiance) values influenced by cloud cover.

The Cloud Index is a quantitative value—usually between 0 and 1 or occasionally greater than 1—that indicates the degree to which the sun's irradiance is obstructed or scattered by

cloud cover. It is an indicator of atmospheric clarity, where:

- CI ≈ 0 represents a completely overcast sky (heavy cloud cover),
- CI ≈ 1 represents a clear sky with little or no cloud interference
- CI > 1 is possible in exceptional circumstances because of cloud edge enhancements (over irradiance), although in reality, it's typically limited.

Steps involved in Cloud Index Computation

1. Cloud Mask Aggregation

- The binary or probabilistic cloud mask derived from the autoencoder is examined over the region of interest.
- Each pixel is given a cloudiness value (0 = clear, 1 = cloud).
- Total cloud cover in an area is calculated as the average of cloud mask for all pixels:

$$\textbf{Cloud Fraction} = 1/N \sum_{i=1}^{N} \textbf{Cloud Mask}(i)$$

equation iii

where N represents the total pixels contained in the chosen region.

2. Obtaining Cloud Index from Cloud Fraction

- A simple mapping function can be used to convert cloud fraction into a cloud index. For instance:

$$\textbf{CI} = 1 - \alpha \cdot \textbf{Cloud Fraction}$$

equation iv

where $\alpha \in [0.8,1.2]$ is a scaling factor calibrated from past GHI measurements. This would imply that an area with 40% cloud cover could have a CI of 0.6, lowering the theoretical GHI proportionally.

Application in GHI Estimation

Cloud index is utilized to modulate clear-sky irradiance computed through physical solar models such as those in PVLib. The formula is:

$$\textbf{Estimated GHI} = \textbf{Clear Sky GHI} \times \textbf{Cloud Index}$$

equation v

This enables the model to include real-time atmospheric fluctuation into the GHI estimation, hence generating a more realistic irradiance profile than clear-sky models by themselves.

Significance of Cloud Index

- Enables location-time and time-location specific adjustments to be made to GHI values.
- Enables real-time solar forecasting, even in regions without ground-based irradiance sensors.
- Captures dynamic atmospheric phenomena such as moving cloud shadows, thick overcast layers, or sudden clearances.
- Serves as an intermediary output that also serves to be visualized and interpreted for tracking sky conditions.

3.5.3 INITIAL GHI ESTIMATION USING PVLIB

The initial GHI estimate in this project is computed using PVLib, a Python library for solar energy modeling. PVLib uses physical models to simulate clear-sky GHI based on location data (latitude, longitude, elevation) and timestamps aligned with satellite images. It calculates solar positions and then uses models like Ineichen or Haurwitz to estimate GHI under cloud-free conditions.

To account for cloud cover, the clear-sky GHI is scaled using a Cloud Index derived from satellite data, producing a more realistic GHI estimate. This adjusted GHI is then used as input for ConvLSTM-based forecasting. PVLib ensures physical accuracy and serves as a reliable baseline for evaluating the deep learning model.

3.5.4 DATA FLOW

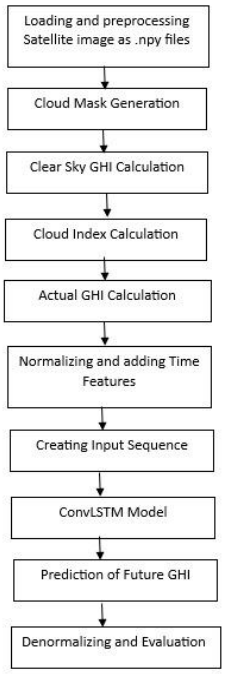

Figure 3.5: Data Flow Diagram

3.6 AUTOENCODER BASED SPATIOTEMPORAL MODELING

The core of the GHI prediction system is an autoencoder-based spatiotemporal model that captures spatial and temporal patterns in satellite image sequences. The encoder uses convolutional layers to extract spatial features like cloud edges and densities, while ConvLSTM units preserve temporal dependencies. This results in a compressed latent representation that summarizes key spatiotemporal features, reducing data size while retaining essential dynamics.

The decoder reconstructs future satellite frames from this latent space using ConvLSTM and upsampling layers. It maintains temporal continuity, modeling cloud motion realistically. These predicted frames resemble real satellite images and are used to estimate future cloud cover and irradiance values for GHI forecasting.

3.7 MODEL LAYERS AND HYPERPARAMETERS

The proposed spatiotemporal autoencoder model for GHI forecasting is built on a structured sequence of layers designed to handle complex spatiotemporal patterns from satellite images. The model begins with Conv2D blocks that extract low- and high-level spatial features such as edges, gradients, and cloud boundaries from each frame. These are typically followed by batch normalization and pooling layers for regularization and spatial downsampling, aiding in efficient training and abstraction.

To capture temporal dynamics, the model integrates ConvLSTM layers**,** which extend LSTM cells with convolutional operations. This enables the network to model both spatial and temporal dependencies, particularly the movement of clouds across frames, essential for accurate GHI prediction. ConvLSTM units maintain memory across time steps, allowing the network to learn evolving cloud patterns and irradiance trends.

The model architecture includes an encoder with Conv2D and pooling layers, followed by ConvLSTM blocks forming the temporal bottleneck. The decoder mirrors the encoder using upsampling or transposed convolutions to reconstruct the spatial dimensions of the image sequence. The model summary outlines layer types, output shapes, and trainable parameters—ranging from hundreds of thousands to millions—depending on image resolution and network depth.

3.8 TRAINING STRATEGY AND LOSS FUNCTIONS

The training strategy chosen for this project is thoughtfully designed to allow the spatiotemporal autoencoder to effectively learn about the temporal and spatial information inherent in visible satellite imagery. The intention is to make sure that the model should be able to reconstruct useful representations of cloud motion and precisely estimate Global Horizontal Irradiance (GHI) values.

Mean Squared Error (MSE) is selected as the main loss function and adaptable to image

reconstruction tasks. MSE is computed as the mean of the squared differences between the estimated pixel intensity and the associated ground truth pixel intensity.

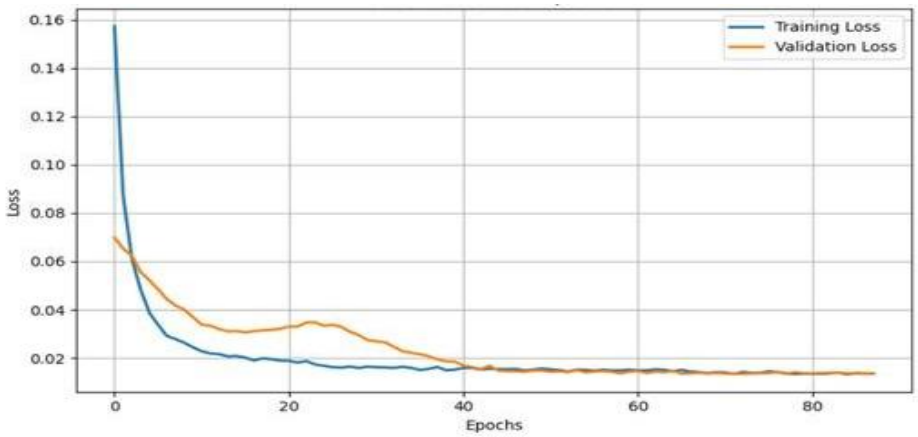

Figure 3.6: Model Loss Over Epochs

The training vs validation loss curve in figure 3.6 demonstrates stable convergence with no signs of overfitting. Both losses decrease steadily and plateau around 0.02, indicating that the model has effectively learned from the training data and generalizes well on unseen validation samples. This supports the model's reliability in predicting GHI from satellite imagery.

3.9 GHI ESTIMATION FROM RECONSTRUCTED FRAMES

The final and most crucial step in the GHI forecasting pipeline is estimating Global Horizontal Irradiance from the reconstructed satellite image frames. This step converts visual cloud data into actionable meteorological information, key for renewable energy applications. The reconstructed frames, which retain spatial and temporal cloud dynamics, undergo post-processing using cloud index (CI) computation and irradiance scaling to derive pixel-level GHI values.

The cloud index serves as a bridge between satellite-derived brightness and atmospheric transparency, directly impacting GHI. CI is calculated for each pixel by linearly normalizing its brightness between two thresholds: B_clear (brightest, clear skies) and B_cloudy (darkest, dense clouds). CI values range from 0 (clear sky) to 1 (full cloud cover),

forming a spatial map of atmospheric opacity. This CI map is then used to estimate surface solar radiation, completing the transformation from imagery to quantitative irradiance data.

$$\boldsymbol{Actual\ GHI\ = Clear\ Sky\ GHI\ \times Cloud\ Index}$$

equation vi

Once the cloud index (CI) map is generated, it is used with empirical models or regression equations to compute GHI values. These models relate CI to GHI based on factors like solar zenith angle and time of day. In this project, a CI-to-GHI regression based on historical data maps lower CI (clear skies) to higher GHI and higher CI to lower irradiance, aligning with atmospheric science principles.

To enhance stability and performance, the estimated GHI values are scaled and normalized. Normalization (0 to 1 range) simplifies visualization and analysis, while for practical applications like energy forecasting, these values are remapped to physical units (W/m^2) using min-max values from the training data. This two-way normalization ensures both model flexibility and real-world applicability.

3.10 MODEL OPTIMIZATION AND VALIDATION

The main goal of model optimization in our research was to minimize prediction error, ensure stable performance under different atmospheric conditions, and improve the generalizability of learned patterns. In order to accomplish this, a comprehensive plan was devised that entailed tracking training and validation loss curves, detecting and mitigating overfitting behavior, and hyperparameter fine-tuning through several experimental configurations.

During the training period, both training and validation loss were monitored at all times with the Mean Squared Error (MSE) loss function. This was to detect differences in how well the model learned from the training data set compared to how well it performed on the validation set, data that was unseen during training

In an attempt to combat overfitting, we integrated a number of architectural and training-level interventions. We added dropout layers after critical layers in the encoder-decoder and ConvLSTM blocks. Dropout layers randomly disengage a set of neurons within each iteration, essentially compelling the network to discover redundant and unique features instead of relying on distinct neuron pathways. An empirical 0.3-0.5 dropout rate was determined to offer optimal regularization. Furthermore, batch normalization layers were incorporated to normalize intermediate activations, which stabilized the learning process by reducing internal covariate shift and contributed to speeding up convergence without requiring very small learning rates.

Early stopping was also utilized while training. The validation loss should not improve if the training was stopped after some epochs. Training was automatically stopped, and the model was restored to the best-performing state captured up to that point. This avoided overtraining the model and kept only the version with best generalization. Additionally, L2 regularization, or weight decay, was implemented on convolutional and dense layers, penalizing large weights and forcing the model to converge towards simpler, more generalizable representations. This method effectively simplified the model and decreased the likelihood of overfitting.

Adam was employed as the optimizer during training due to its flexibility and computational efficiency. We tuned its parameters—like the beta values that regulate the decay rates of moving averages and the epsilon term employed for numerical stability—to suit the nature of our dataset better. During training, model checkpointing was done to store the weights after each epoch if validation loss improved. This helped keep the best version of the model at hand, even if later epochs compromised its performance.

Chapter 4

Implementation

4.1 TOOLS AND LIBRARIES USED

The design of the spatiotemporal autoencoder-based system for predicting GHI depends extensively upon a strong group of tools and libraries, which are primarily centered on the Python programming environment. Python is selected because of its ease, extensive community support, and the presence of strong libraries optimized for deep learning, image manipulation, and data analysis.

- For training and building deep learning models, TensorFlow and Keras are the main frameworks.
- For handling and extracting data from HDF5 files, the h5py package is employed.
- For preprocessing and data manipulation, Pandas and NumPy are essential. NumPy enables fast numerical computations and array manipulations, whereas Pandas is perfect for dealing with tabular metadata and time-series image references.
- Matplotlib is utilized extensively to visualize image sequences, loss curves, cloud masks, and GHI predictions. These plots are crucial for model behavior and performance interpretation.

4.2 DATASET LOADING AND SEQUENCE GENERATION

In deep learning-based solar irradiance forecasting using satellite images, the dataset loading and sequence generation pipeline plays a vital role. It transforms raw satellite data into a format suitable for ConvLSTM models, preserving both spatial and temporal patterns. Typically, satellite images arrive as TIFF raster data captured at fixed intervals. These frames undergo preprocessing—often with an autoencoder—for denoising and dimensionality reduction. The cleaned images are then chronologically ordered and grouped into sequences, forming 3D or 4D arrays required for ConvLSTM input. This

involves generating overlapping or non-overlapping batches of consecutive frames to maintain temporal continuity for model training.

4.2.1 SAMPLE FRAME GENERATION

Sample frame generation is the first step in creating structured datasets from reconstructed satellite images. Each 2D frame, resized to 64×64 pixels, is saved as a .npy file for efficient processing. Choosing the right resolution balances detail and computation—too high increases training time, too low loses key cloud features. Using a sliding window approach, frames are chronologically ordered by timestamps to preserve temporal consistency. Corrupted or misaligned frames are filtered out to maintain data quality. Valid frames are then paired with GHI values using timestamps, forming the final dataset for training.

4.2.2 TEMPORAL BATCHING

Temporal batching is the most critical part of sequence generation in time-series models. It converts the independent spatial frames to time-aware spatiotemporal blocks capable of detecting both motion and transformation in cloud structures over time. Temporal batching simply groups frames into fixed-size sequences, which are a sequence of consecutive images taken at fixed time intervals (e.g., every 30 minutes). For instance, suppose we wish to construct sequences of 3 frames each, and our dataset contains images taken every half an hour between 6:00 AM to 6:00 PM. We would create sequences like:

Sequence 1: 6:00 AM, 6:30 AM, 7:00 AM

Sequence 2: 6:30 AM, 7:00 AM, 7:30 AM

…

and so on.

This sliding window technique aids in creating a dense dataset with overlapping temporal data. The benefit of overlapping sequences is that the model has an opportunity to learn transitions and cloud movement patterns more often, which is useful under dynamic

weather. Each sequence, once formed, becomes a 3D matrix (for grayscale) or a 4D tensor (for RGB), of size [timesteps, height, width] or [timesteps, height, width, channels].

Target value assignment is another aspect of batching. For every input frame sequence, we need to assign a ground truth label — the predicted GHI value. The GHI value is usually measured at some future point in time, 2 hour from the end of the last frame in the input sequence. This makes the model learn to "forecast ahead" from current and past cloud patterns.

4.3 MODEL DEVELOPMENT AND TRAINING

Development and training of the spatiotemporal autoencoder model using satellite images consists of a well-defined pipeline that can process large sets of image data efficiently while allowing the model to learn valid temporal and spatial patterns applicable for GHI forecasting.

Model Instantiation and Training Loop

The first one is the instantiation of the model, where the architecture—made up of the convolutional encoder, ConvLSTM temporal processor, and decoder—is instantiated with TensorFlow and Keras. The model is initialized with proper input dimensions depending on the spatiotemporal cubes (e.g., sequences of 6 gray-scale images of size 256×256). Each layer is defined meticulously in order to retain spatial information as well as learn temporal dependencies.

After the model is initialized, the training loop is started. Satellite image sequences are loaded in batches and passed into the model, where the encoder reduces them into feature-dense representations. These representations are passed through the ConvLSTM layers to learn temporal dynamics before decoding into the next frame prediction.

4.4 CLOUD INDEX CALCULATION AND INTEGRATION

The computation of solar irradiance from satellite imagery depends strongly on knowing and measuring cloud cover. The topic of this section is how information on clouds is obtained from satellite imagery and incorporated into the GHI prediction scheme. Two of the essential subelements in this chain are computing the cloud mask from brightness thresholding, and deriving the cloud index and its relationship to GHI.

Brightness thresholding is a common technique in satellite image processing to detect cloud-covered areas. As clouds are usually brighter in visible channel satellite images (because of high reflectance), a threshold value can be set to separate clouds from the background (i.e., clear-sky or land surface).

$$\boldsymbol{Cloud\ Index} = \mathbf{1} - \frac{\boldsymbol{N} - \boldsymbol{N_{min}}}{\boldsymbol{N_{max}} - \boldsymbol{N_{min}}}$$

equation vii

In this method, every pixel in the grayscale satellite image is compared with a pre-set brightness threshold. If the pixel intensity is above this threshold, it is classified as a cloud pixel; otherwise, it is treated as clear sky. This process yields a binary cloud mask such that '1' indicates cloudy regions and '0' indicates clear regions. Cloud mask is an important intermediate representation that is propagated to the subsequent phases of the GHI approximation procedure.

4.5 VISUALIZING INTERMEDIATE AND FINAL OUTPUTS

Input sequences to the spatiotemporal autoencoder model are grayscale satellite imagery taken at a regular time interval (30 minutes). These sequences are conjoined to create spatiotemporal cubes with a fixed number of frames—typically between 6 and 8. Inputs supply temporal context for the model to learn cloud motion and development trends. Such frames are usually shown with the ground-truth next frame for visual comparison. Differs, particularly in cloud borders and brightness, mark the reconstruction's weaknesses or strengths.

Final Ghi

The final product of the whole pipeline is the forecasted GHI map, indicating the spatial variation of solar irradiance over the region at the predicted time. The maps are calculated from the forecasted satellite image, cloud mask, cloud index, and clear-sky irradiance models of PVlib. Figure 4.1 and the figure 4.2 shows the actual vs predicted GHI value of first September 2019 11-2:30 am. From this we can conclude that the prediction is accurate.

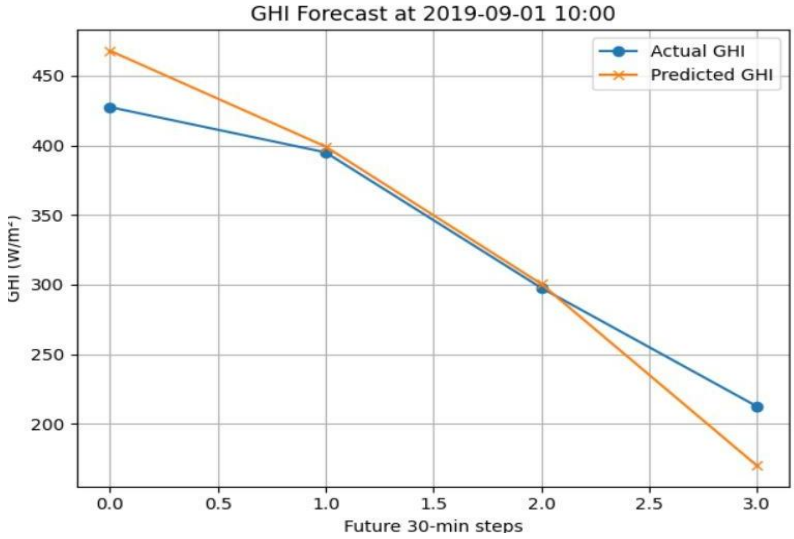

Figure 4.1: GHI prediction

Timestamp	Actual GHI (W/m²)	Predicted GHI (W/m²)
2019-09-01 11:00:00	730.685730	654.868958
2019-09-01 11:30:00	738.004333	614.037720
2019-09-01 12:00:00	753.821838	687.991638
2019-09-01 12:30:00	495.192383	491.924835
2019-09-01 13:00:00	610.296082	617.863220
2019-09-01 13:30:00	483.168793	518.268982
2019-09-01 14:00:00	636.951965	545.025269
2019-09-01 14:30:00	573.709534	507.457520
2019-09-01 15:00:00	552.803955	507.118835
2019-09-01 15:30:00	427.424500	402.269714

Figure 4.2 Actual vs Predicted GHI Values

Chapter 5

Results

5.1 EVALUATION METRICS

For testing the precision and efficacy of the suggested cloud identification and GHI labeling model, three core metrics of assessment used are Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Squared Error (MSE). The same are well employed in the fields of regression as well as highly important while reviewing the way well the model generalizes towards estimations for solar irradiance levels at the surface based on deriving cloud masks using satellite imagery.

5.1.1 ROOT MEAN SQUARE ERROR (RMSE)

- RMSE penalizes larger errors more than MAE. A score of 173.90 indicates some larger prediction errors exist in the dataset.
- The presence of larger prediction deviations might be due to sudden cloud changes, edge cases, or noise in satellite input.

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

equation viii

5.1.2 MEAN ABSOLUTE ERROR (MAE)

- Predicted GHI values deviate by about 128.81 W/m² from the actual values.
- For GHI data this is a moderate error.
- A lower MAE (closer to 0) would suggest highly accurate predictions. 128.81 suggests that the model performs decently
- The equation for MAE is:

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

equation ix

5.1.3 MEAN SQUARED ERROR (MSE)

MSE is the mean of the squared differences between the predicted and actual values. this is used to evaluate the performance of regression models, particularly to determine how spread out the error is. MSE helps in determining the level of deviation in GHI predictions based on any errors in cloud mask generation or model learning. The formula is as follows:

$$\boldsymbol{MSE} = \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}(\boldsymbol{y_i} - \boldsymbol{\hat{y}_i})^{\mathbf{2}}$$

equation x

5.1.4 R^2 SCORE

- This score tells much variance in actual GHI can the model can explain. A score of 0.67 means 67% of the variation in GHI is captured by the model.
- R^2 values range from 0 (no explanation of variance) to 1 (perfect explanation).
- An R^2 of 0.67 is good level of accuracy. It shows the model is learning meaningful patterns. With more data diversity, feature enrichment, or advanced modeling, the score can be improved if needed.

5.2 PREDICTION ACCURACY AND VISUALIZATION

In this research, accuracy of prediction is not only evaluated using numerical measures but also by graphical comparison techniques such as Ground Truth vs. Predicted GHI (Global Horizontal Irradiance) maps and training/validation loss curve plots. These visualizations give more in-depth insights into how well the model has learnt the patterns within the data and where it prefers to go astray.

Table 5.1 shows the evaluation matrix. The R^2 (R-squared), RMSE and MSE according the model output.

Evaluation Metrics	**Score**
MAE (Mean Absolute Error)	128.81
RMSE (Root Mean Squared Error)	173.90
R^2 (R-squared)	0.67

Table 5.1: Evaluation Metrics

5.2.1 GROUND TRUTH VS PREDICTED GHI

One of the most insightful methods to assess model performance in spatial prediction tasks is to compare the actual (ground truth) and predicted outputs side-by-side as heatmaps or geographical maps. In our project, GHI maps are derived from the satellite image-based cloud masks and compared against the ground truth GHI values obtained from the Tirupati ground station data.

These maps give a pixel-by-pixel visual contrast of irradiance intensity levels over an area, enabling an evaluation of the spatial correspondence between predicted and true values. The areas with high solar irradiance are represented lighter, whereas the areas with clouds and low irradiance are represented darker. Through visual comparison of both the maps for a particular timestamp, we can determine how well the model has been able to simulate the actual solar radiation patterns.

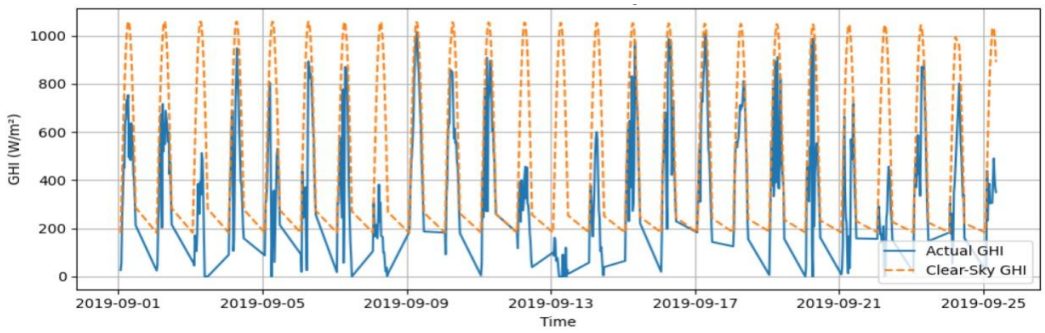

Figure 5.1: Actual vs Clear sky GHI

The figure in 5.1 is the comparison of Actual GHI and Clear-Sky GHI from September 1–25, 2019, highlighting the impact of cloud cover and atmospheric variability on solar irradiance.

5.2.2 LOSS CURVE PLOTS

The training loss curve should generally be decreasing, showing that the model is learning from the training set effectively. A steep decline in the first few epochs followed by a slow flattening is a sign of a well-trained model. The validation loss curve, however, would be used to check for overfitting or underfitting. Both the training and validation losses should

decrease and plateau at about the same level. A large gap between them—particularly if the validation loss rises while training loss still falls—can be a sign of overfitting, in which the model works well on training data but badly on fresh, unseen data.

In this research, the loss curves are employed for choosing the optimal-trained model checkpoint and for fine-tuning parameters like learning rate, batch size, and epochs. Smooth and stable loss curves indicate a successful learning process and strengthen the model's credibility in real-world solar forecasting applications.

These visualizations, in conjunction with evaluation metrics, provide a comprehensive picture of model performance. They confirm the efficacy and accuracy of the proposed approach in predicting solar irradiance using satellite-derived cloud masks, a major achievement in remote sensing-based renewable energy forecasting.

Chapter 6

Discussions and Limitations

6.1 SUMMARY OF FINDINGS

Spatiotemporal Learning Improves GHI Prediction

The combination of autoencoders and ConvLSTM layers was very effective for spatiotemporal pattern modeling that is characteristic of satellite cloud images. The model was able to capture the spatial texture of clouds and their temporal progression, which are decisive factors of solar irradiance. In contrast to conventional frame-by-frame models, this method accounted for image sequences, enabling it to predict cloud motion and forecast irradiance variations more accurately.

GHI Estimation Pipeline Demonstrated to be Accurate and Scalable

The pipeline efficiently converted satellite observations into GHI predictions by combining cloud index values with PVlib's clear-sky irradiance model. The reconstructed satellite frames, upon processing for irradiance estimation and cloud detection, provided GHI maps that showed consistency and reliability. The modular structure allows the system to be scaled for larger areas easily or expanded with multispectral data in the future.

Visualization Demonstrated Model Interpretability

Visualizations of input image sequences, autoencoder reconstructions, and resulting GHI maps provided robust qualitative demonstrations of the model's ability to learn. Predicted frames were similar to actual satellite imagery, and generated GHI maps exhibited realistic solar irradiance gradients and variability.

Performance Metrics Showed Robust Training

The model had low reconstruction loss (MSE) and worked well against assessment metrics like RMSE, MAE, and MSE when predicted GHI values were compared with baseline or true values. These findings reinforce the efficacy of training approach, including utilization of the Adam optimizer, judiciously selected hyperparameters, and early stopping/checkpointing methodologies.

6.2 CHALLENGES IN CLOUD MODELING AND GHI PREDICTION

Cloud behavior modeling and precise prediction of Global Horizontal Irradiance (GHI) from satellite images pose a variety of technical and practical challenges. These challenges cut across data quality, model limitations, and the natural variability of atmospheric conditions. The major difficulties faced in this project are discussed below:

- High Variability and Dynamics of Cloud Behavior: Clouds display irregular, nonlinear, and strongly dynamic motion patterns that vary strongly with time. They vary in density, elevation, texture, and extent, so it is problematic to make generalizations about their effect on solar irradiance. Small errors in interpreting the thickness or motion of clouds can cause large errors in GHI estimation. Representing such variability using lowbandwidth satellite imagery is inherently difficult and involves highfidelity temporal modeling.
- Restricted Spectral Resolution of Visible Satellite Images: The work makes use of visible light satellite images (grayscale) for cloud identification. Such images do not contain infrared or thermal information, which may also reveal more on cloud height, temperature, and type. Consequently, valuable information regarding vertical cloud shapes or multilayer cloud systems is concealed, possibly decreasing the accuracy of the cloud mask and the derived GHI forecast.
- Sensitivity of Brightness Thresholding: The brightness based cloud masking method, while straightforward and effective, is sensitive to changes in surface reflectance, time of day, and sun satellite angle. For example, bright surfaces such as deserts or snowy areas could be incorrectly identified as cloud cover. In the same manner, thin or transparent clouds may not be well differentiated, impacting the subsequent GHI estimation.
- Temporal Alignment and Data Gaps: Preserving the temporal consistency of satellite images is essential for capturing temporal dependencies in modeling. But actual satellite data can be affected by missing frames, uneven intervals, or artifacts caused by atmospheric noise. Such inconsistencies pose challenges in generating clean input sequences for the ConvLSTM layers, which need temporally aligned inputs for effective learning.

- Lack of Ground Truth Data for Supervision: Ground Based pyranometer measurements are scarce and frequently missing in distant or underdeveloped areas. That restricts the availability of labeled datasets for model training and validation of the GHI predictions. In the absence of extensive ground truth, it is hard to judge the model's actual performance or tune it towards regional differences in irradiance behavior.
- Overfitting and Generalization Risks: Deep models with large numbers of parameters, e.g., autoencoders and ConvLSTM networks, are susceptible to overfitting, particularly when the models are trained on small or very specific datasets. To guarantee the model will generalize well to unseen areas, other seasons, or diverse climatic zones, proper regularization, data augmentation, and validation plans are necessary.

6.3 IMPACT OF IMAGE RESOLUTION AND DATA AVAILABILITY

Accuracy and resilience of Global Horizontal Irradiance (GHI) forecasting heavily rely on image resolution and data consistency, high quality. Herein, how both aspects contribute to model performance, quality of predictions, and the overall framework applicability is elaborated.

Impact of Image Resolution on Feature Extraction

High-resolution satellite imagery enables the model to identify fine-grained cloud features like edges, tiny cumulus clouds, and local discontinuities in cloud cover. These characteristics are essential for comprehending irradiance variations locally. At lower resolutions, however, cloud boundaries become blurry, perhaps missing important transitions that affect GHI.

- An example is a resolution of 256×256 pixels being adequate for regional forecasting but potentially missing localized cloud shadows or thin clouds.
- Increased resolution improves spatial granularity but at increased computational cost in terms of memory and processing time.
- The model can also need additional training data and deeper networks to take advantage of the additional information provided by higher resolution.

Constraints Due to Availability of Data

Satellite images are available at fixed time intervals (e.g., 10–30 minutes), but full temporal coverage is not always possible because of hardware failure or transmission delays, Atmospheric distortions or sensor faults and calibration problems. These gaps in data can disrupt sequence modeling, which is particularly troublesome for ConvLSTM-based models that heavily depend on temporal continuity. Loss of frames can result in improper learning of temporal patterns, reflecting on the accuracy of GHI predictions.

Regional and Seasonal Data Bias

The performance of the model is also related to training data diversity. If the satellite images predominantly utilized are from a particular region or season, the model may learn region-specific or climate-specific biases and perform poorly under other conditions.

- For instance, cloud behavior over the Indian subcontinent during the monsoon season is vastly different from dry regions in the Middle East or snowy areas in Europe.
- Thus, the presence of various training data throughout the seasons and geography is essential for a model to be trustworthy in actual use.

Supervised Learning Ground Truth Availability

Although satellite information is rich in volume, ground truth readings of GHI (such as from pyranometers) are sparse and scarce. This impacts the capability to train supervised models, cross-validate predictions, and fine-tune for precision improvement.

Without labeled data, the system will have to use proxy techniques such as GHI estimation based on cloud indices or synthetic data, which can incur systematic biases.

Requirement of Data Harmonization and Preprocessing

Availability of data is usually hampered by non-uniform formats, varying resolutions, and different calibration methods between satellites and time intervals. Harmonizing such data prior to inputting it into the model requires extensive preprocessing—e.g., clipping areas,

brightness normalization, grayscale conversion, etc. Model inputs can become inconsistent if not harmonized properly, impacting both learning and inference phases.

6.4 LIMITATIONS OF THE AUTOENCODER FRAMEWORK

Though it performs well in learning spatiotemporal patterns from satellite imagery, the autoencoder framework adopted for GHI prediction has a number of inherent limitations. The architectural and functional limitations of the framework need to be recognized to help drive future improvements. Some of the major limitations are discussed point-wise as follows:

Loss of Fine Spatial Details During Compression

- Autoencoders downsample the input image by convolution, which may discard fine-grained spatial details.
- Significant cloud edges, thin cloud sheets, or small brightness changes that influence irradiance might be smeared or lost entirely during encoding.

Reconstruction Focus Rather Than Direct Prediction

- The main objective of an autoencoder is to reconstruct the input rather than directly predict GHI.
- This indirect method might not always yield the most physically meaningful features for irradiance estimation, as opposed to models specifically trained to regress GHI values.

High Dependency on Quality of Input Sequences

- The model is very sensitive to the temporal coherence and quality of input satellite frames.
- Missing frames, low-resolution data, or inconsistent lighting conditions can lead to serious degradation in reconstruction accuracy, impacting the downstream GHI estimation.

Conclusions and Future Work

Enhancing Spatial And Temporal Modeling

To further enhance the accuracy and reliability of GHI prediction models, improving spatial and temporal modeling performance is an imperative area of future research. Although the existing architecture is efficient, it can be enhanced in some ways to extract the complexity in atmospheric patterns as well as in their temporal progression.

Addition of Multi-Scale Feature Extraction

- Basic convolutional layers have one of the drawbacks of extracting features only at a fixed spatial scale.
- Using multi-scale convolutional filters or feature pyramid networks (FPNs) can assist the model in perceiving fine-grained and large global spatial patterns, e.g., small groups of clouds and global weather regimes.

Application of Attention Mechanisms

- Incorporating spatial and temporal attention mechanisms enables the model to pay greater attention to important areas and time steps that play major roles in GHI variations.
- For example, attention layers can allow the model to focus on areas with dense or high-speed clouds, enhancing both interpretability and prediction performance.

Higher-Order Temporal Modeling using Transformers

- Although ConvLSTM is highly effective in modeling short-term temporal dependencies, it might not necessarily understand long-range temporal correlations well.
- Substituting or supplementing ConvLSTM layers with temporal transformers or temporal convolutional networks (TCNs) can assist in modeling longer sequences and learning patterns over longer intervals.

Higher Temporal Resolution

- Rather than being trained on satellite images sampled at 30 minutes or so, the system might be trained on higher-frequency image sequences.
- This allows the model to more accurately capture fast cloud dynamics, particularly

useful for nowcasting (very short-term forecasting).

Integration of Optical Flow for Motion Tracking

- Methods such as optical flow can be integrated into the spatial modeling pipeline to track true cloud movement vectors between frames.
- These motion characteristics can be combined with autoencoder outputs to offer more physically relevant temporal development, assisting in precise forecasting of cloud location and thus irradiance.

Residual and Dense Connections

- Deep networks can experience vanishing gradients in very deep models.
- Applying residual blocks (ResNet-type) or dense connections (DenseNet) in encoder and decoder can enhance gradient flow, allowing deeper and more powerful models for spatial pattern extraction.

Spatial Contextual Perception

- Movement and formation of clouds are determined by topography and surrounding areas.
- Incorporating context-aware convolutional modules or local environment features-based region-specific encodings (e.g., surface albedo, elevation) can aid in adapting predictions on the basis of local environmental characteristics.

Dynamic Handling of Sequence Length

- Fixed-length input sequences (e.g., 6 satellite frames) are used at the moment. Future systems should be able to dynamically handle variable-length sequences.
- Such adaptability is vital for practical implementations where data availability can change and sequence length can be different.

3D Convolutional Techniques

- Substituting the use of merely 2D CNNs and temporal modules with 3D CNNs (convolution across space and time) could deliver a single technique for spatiotemporal modeling.
- 3D convolutions in some instances allow for more effective extraction of joint spatiotemporal features, especially with short sequences.

CONCLUSION

The work effectively showcases an autoencoder based on deep learning for forecasting Global Horizontal Irradiance (GHI) via visible satellite observations. Through its utilization of spatiotemporal autoencoders and the integration of ConvLSTM layers, the system captures spatial trends and temporal features of cloud displacements—both key elements involved in precise forecastings of solar irradiance.

Application of INSAT satellite imagery guarantees wide spatial coverage and regular updates, overcoming the drawbacks of conventional ground-based pyranometers that provide localized and frequently delayed data. With optimized data preprocessing, clipping of regions, masking of clouds, and incorporation of PVLib for baseline GHI estimation, the system can predict irradiance based on the behavior of clouds, even for locations with no on-ground sensors.

Training techniques like the application of MSE loss, the Adam optimizer, and hyperparameter tuning with caution enabled the model to converge well and make accurate predictions. Visualizations of intermediate frames and reconstructed outputs confirm the model's comprehension of cloud transitions over time.

Although encouraging results, the framework also has its limitations in terms of sensitivity to image resolution, requirement of region-specific data for generalization, and the approximations used during cloud index estimation. Yet, with future developments—such as the addition of multi-spectral channels, higher resolution inputs, and scaling to multiple geographic regions—the framework can be further reinforced.

References

[1] L. Zou, Ping Tang, Yisen Niu, Zixuan Yan, Xilong Lin, Jifeng Song , and Qian Wang “A Cloud Motion Estimation Method Based on Cloud Image Depth Feature Matching,” IEEE Geosci. Remote Sensing Lett., vol. 22, pp. 1–5, 2025, doi: 10.1109/LGRS.2024.3491094.

[2] Y. Son, Y. Yoon, J. Cho, and S. Choi, “Cloud Cover Forecast Based on Correlation Analysis on Satellite Images for Short-Term Photovoltaic Power Forecasting,” Sustainability, vol. 14, no. 8, p. 4427, Apr. 2022, doi: 10.3390/su14084427.

[3] M. Perera, J. D. Hoog, K. Bandara, and S. Halgamuge, “Distributed solar generation forecasting using attention-based deep neural networks for cloud movement prediction,” Nov. 17, 2024, arXiv: arXiv:2411.10921. doi: 10.48550/arXiv.2411.10921.

[4] D. Yu, S. Lee, S. Lee, W. Choi, and L. Liu, “Forecasting Photovoltaic Power Generation Using Satellite Images,” Energies, vol. 13, no. 24, p. 6603, Dec. 2020, doi: 10.3390/en13246603.

[5] C. K. Kim, H.-G. Kim, Y.-H. Kang, C.-Y. Yun, and Y. G. Lee, “Intercomparison of Satellite-Derived Solar Irradiance from the GEO-KOMSAT-2A and HIMAWARI-8/9 Satellites by the Evaluation with Ground Observations,” Remote Sensing, vol. 12, no. 13, p. 2149, Jul. 2020, doi: 10.3390/rs12132149.

[6] Y. Son, X. Zhang, Y. Yoon, J. Cho, and S. Choi, “LSTM–GAN based cloud movement prediction in satellite images for PV forecast,” J Ambient Intell Human Comput, vol. 14, no. 9, pp. 12373–12386, Sep. 2023, doi: 10.1007/s12652-022-04333-7.

[7] P. Kosmopoulos, Harshal Dhake b, Nefeli Melita c, Konstantinos Tagarakis d, Aggelos Georgakis d, Avgoustinos Stefas e, Orestis Vaggelis c, Valentina Korre d, Yashwant Kashyap "Multi-Layer Cloud Motion Vector Forecasting for Solar Energy Applications," Applied Energy, vol. 353, p. 122144, Jan. 2024, doi: 10.1016/j.apenergy.2023.122144.

[8] A. A. Prasad and M. Kay, "Prediction of Solar Power Using Near-Real Time Satellite Data," Energies, vol. 14, no. 18, p. 5865, Sep. 2021, doi: 10.3390/en14185865.

[9] R. Alonso-Suarez, F. Marchesoni, L. Dovat, and A. Laguarda, "Satellite-Based Operational Solar Irradiance Forecast for Uruguay's Solar Power Plants," in 2021 IEEE URUCON, Montevideo, Uruguay: IEEE, Nov. 2021, pp. 182–187. doi: 10.1109/URUCON53396.2021.9647087.

[10] N. Straub, S. Karalus, W. Herzberg, and E. Lorenz, "Satellite - Based Solar Irradiance Forecasting: Replacing Cloud Motion Vectors by Deep Learning," Solar RRL, vol. 8, no. 24, p. 2400475, Dec. 2024, doi: 10.1002/solr.202400475.

[11] A. S. Devi, G. Maragatham, K. Boopathi, and M. R. Prabu, "Short-term solar power forecasting using satellite images".

[12] H. S. Jang, K. Y. Bae, H.-S. Park, and D. K. Sung, "Solar Power Prediction Based on Satellite Images and Support Vector Machine," IEEE Trans. Sustain. Energy, vol. 7, no. 3, pp. 1255–1263, Jul. 2016, doi: 10.1109/TSTE.2016.2535466.

[13] J. Polo L.F. Zarzalejo a, M. Cony b, A.A. Navarro a, R. Marchante b, L. Martı ´ n b, M. Romero, "Solar radiation estimations over India using Meteosat satellite images," Solar Energy, vol. 85, no. 9, pp. 2395–2406, Sep. 2011, doi: 10.1016/j.solener.2011.07.004.

[14] J. Polo L.F. Zarzalejo a, M. Cony b, A.A. Navarro a, R. Marchante b, L. Martı ´ n b, M. Romero., “Solar radiation estimations over India using Meteosat satellite images,” Solar Energy, vol. 85, no. 9, pp. 2395–2406, Sep. 2011, doi: 10.1016/j.solener.2011.07.004.

[15] H. G. Kamath and J. Srinivasan, “Validation of global irradiance derived from INSAT-3D over India,” Solar Energy, vol. 202, pp. 45–54, May 2020, doi: 10.1016/j.solener.2020.03.084.

Appendices

```python
# Load & Extract INSAT Data
input_directory = "/content/drive/MyDrive/solar_project/2019_Data/"
output_directory = "/content/drive/MyDrive/solar_project/p1"


def load_insat_data(hdf5_file, dataset_name):
    with h5py.File(hdf5_file, 'r') as f:
        if dataset_name not in f:
            return None
        return f[dataset_name][:]

def process_hdf5(file_path, dataset_name):
    data = load_insat_data(file_path, dataset_name)
    return data if data is not None else None

if not os.path.exists(output_directory):
    os.makedirs(output_directory)

h5_files = [f for f in os.listdir(input_directory) if f.endswith(".h5")]

dataset_name = "IMG_VIS"  # Adjust based on the INSAT product
```

Figure 7.1: Loading Data

```python
# Define Coordinates
lat, lon = 13.627, 79.397
npy_files = [f for f in os.listdir(output_directory) if f.endswith(".npy")]
for npy_file in npy_files:
    npy_path = os.path.join(output_directory, npy_file)
    insat_data = np.load(npy_path)
    clipped_insat_images = extract_insat_region(insat_data, lat, lon)
    num_frames, height, width = clipped_insat_images.shape
    clipped_insat_images = clipped_insat_images.reshape((num_frames, height, width, 1))
    output_file = os.path.join(Clipped_directory, npy_file)
    np.save(output_file, clipped_insat_images)
    print(f"Saved clipped images to {output_file}")
```

Figure 7.2: Clip AOI

```
latent_dim = 64
epochs = 10
batch_size = 4

def build_autoencoder(input_shape):
    inputs = Input(shape=input_shape)

    # Encoder
    x = Conv2D(64, (3, 3), activation='relu', padding='same')(inputs)
    x = Conv2D(32, (3, 3), activation='relu', padding='same')(x)
    x = Conv2D(16, (3, 3), activation='relu', padding='same')(x)
    encoded = Flatten()(x)
    latent_space = Dense(latent_dim, activation='relu')(encoded)

    # Decoder
    x = Dense(input_shape[0] * input_shape[1] * 16, activation='relu')(latent_space)
    x = Reshape((input_shape[0], input_shape[1], 16))(x)
    x = Conv2DTranspose(32, (3, 3), activation='relu', padding='same')(x)
    x = Conv2DTranspose(64, (3, 3), activation='relu', padding='same')(x)
    decoded = Conv2DTranspose(1, (3, 3), activation='sigmoid', padding='same')(x)

    autoencoder = Model(inputs, decoded)
    autoencoder.compile(optimizer=Adam(), loss='mse')
    return autoencoder


autoencoder = build_autoencoder(input_shape=X.shape[1:])  # Pass the correct input shape
autoencoder.fit(X_train, X_train, epochs=20, batch_size=50, validation_split=0.2)
```

Figure 7.3: Autoencoder framework

```
def get_cloud_mask(original, reconstructed, threshold=0.08):
    error_map = np.abs(original - reconstructed)
    norm_error = (error_map - error_map.min()) / (error_map.max() - error_map.min() + 1e-6)
    mask = (norm_error > threshold).astype(np.uint8)
    return mask.squeeze(), norm_error.squeeze()

for i, img in enumerate(X):
    recon = autoencoder.predict(img[np.newaxis], verbose=0)[0]
    mask, err_map = get_cloud_mask(img, recon)

    mask_path = os.path.join(mask_output_dir, f"cloudmask_{image_files[i].replace('.npy', '')}.npy")
    np.save(mask_path, mask)

    if i < 3:
        plt.figure(figsize=(12, 4))
        plt.subplot(1, 3, 1)
        plt.imshow(img.squeeze(), cmap='gray'); plt.title("Original")
        plt.subplot(1, 3, 2)
        plt.imshow(mask, cmap='gray'); plt.title("Cloud Mask")
        plt.subplot(1, 3, 3)
        overlay = np.stack([img.squeeze()] * 3, axis=-1)
        overlay[mask == 1] = [1, 0, 0]
        plt.imshow(overlay); plt.title("Overlay")
        plt.show()

    print(f"Saved mask: {mask_path}")
```

Figure 7.4: cloud masking