

NATIONAL INSTITUTE OF TECHNOLOGY MEGHALAYA

DEPARTMENT OF ELECTRICAL ENGINEERING



EE382 – Minor Project

Mid-Term Progress Review Report

On the topic

“Efficient Solar Energy Management System Using Weather Forecasting and Battery Management System”

Report submitted by:-

Name : Ayush Raj
Roll No.: B22EE013

Name : Siddhant Purkayastha
Roll No.: B22CS021

Under the supervision of
Dr. Piyush Pratap Singh
(signature)
(Name of project supervisor)

Introduction

This project aims to develop an integrated system that enhances the efficiency and reliability of solar photovoltaic (PV) energy utilization by combining advanced solar power forecasting with intelligent battery management. Leveraging machine learning models such as Bi-LSTM, LSTM, and GRU, the system accurately predicts solar generation patterns influenced by weather variability, thereby addressing challenges like reverse power flow and voltage deviations. Concurrently, it implements predictive techniques for estimating the State of Charge (SOC) and State of Health (SOH) of batteries, facilitating proactive maintenance and optimal energy storage utilization. By analyzing load curves, weather forecasts, and historical data, the system ensures efficient energy distribution, minimizes wastage, and extends power supply duration, particularly benefiting small industries and institutions. The solution emphasizes a data-driven approach to enable cost-effective energy planning and maximize the sustainability of renewable energy integration into existing power systems.

Motivation

The growing need for clean and green energy solutions, along with rising energy costs and uncertainty of traditional grids, motivated us to design an intelligent solar power system. We wanted to offer a platform where industries or institutions can efficiently forecast their energy needs, generate energy from renewable sources, and supply it intelligently through forecasting systems. By combining weather forecasting and battery analysis, our system optimizes energy supply and battery life, improving environmental and economic sustainability.

Machine Learning Model Overview

This section describes different machine learning models used for accurate prediction of solar power and State of Charge of batteries.

Feedforward Neural Network(FFN) :

The Feedforward Neural Network (FFN) is among the simplest yet foundational architectures in the domain of artificial neural networks. In an FFN, data flows in one direction—from the input layer through one or more hidden layers to the output layer—without any cycles or loops. Despite its structural simplicity, FFNs can capture complex nonlinear mappings between input features and target values, making them suitable for many regression problems, including battery State of Charge (SoC) estimation when time-dependence is minimal or has been implicitly encoded in the input.

In the context of SoC prediction, the FFN receives a flattened representation of input time sequences (typically voltage, current, temperature over a window of time) as a single vector. This allows the model to approximate the functional relationship between recent sensor readings and the current SoC value. Unlike recurrent architectures that explicitly model temporal dependencies, the FFN depends entirely on the temporal context embedded within the fixed-length input window.

The basic architecture of the FFN used in this study comprises an input layer that accepts a vector of dimension $d = \text{seqencelength} \times \text{features}$, followed by one or more hidden layers with nonlinear activations (here ReLU), dropout for regularization, and a final linear layer that outputs a scalar SoC prediction.

The operations for a basic two-layer FFN can be expressed as:

$$\mathbf{z}_1 = \text{ReLU}(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) \quad (1)$$

$$\mathbf{z}'_1 = \text{Dropout}(\mathbf{z}_1, p) \quad (2)$$

$$\hat{y} = \mathbf{W}_2 \mathbf{z}'_1 + \mathbf{b}_2 \quad (3)$$

Where:

- $\mathbf{x} \in R^d$ is the flattened input vector,
- $\mathbf{W}_1, \mathbf{W}_2$ and $\mathbf{b}_1, \mathbf{b}_2$ are learnable weight and bias matrices,
- \hat{y} is the predicted SoC,
- Dropout with probability p is applied during training for regularization.

Although FFNs lack memory of previous states (a limitation for modeling time series data), they often serve as a baseline for comparison with more complex models like LSTM or GRU. Their ease of implementation and fast training times make them appealing for applications where quick estimation is preferred and battery behavior is relatively stable or slowly varying. However, due to the inability to remember past information and patterns, FFNs may struggle to generalize over longer discharges or varying usage patterns which depend on time. This motivates the use of recurrent architectures for more accurate SoC estimation under dynamic conditions.

Long-Short Term Memory (LSTM) :

Long Short-Term Memory (LSTM) networks are a specialized type of Recurrent Neural Network (RNN), designed to learn long-term dependencies in sequential data. In traditional RNNs, information flows from one time step to the next via a hidden state. However, they suffer from the vanishing gradient problem during backpropagation through time, making it difficult to capture dependencies over long sequences. The LSTM architecture, introduced by Hochreiter and Schmidhuber, addresses this limitation using memory cells and a gated mechanism that allows the network to learn when to retain or forget information.

Unlike feedforward networks that consider only the current input, LSTM networks take both the current input x_t and the previous hidden state h_{t-1} as inputs, allowing the model to incorporate temporal dependencies. Each LSTM unit contains three gates: the input gate, the forget gate, and the output gate. These gates control the flow of information to and from the cell state c_t , which acts as a memory line connecting across time steps.

The gates are defined as follows:

- Forget Gate: Decides what information to discard from the cell state.
- Input Gate: Determines what new information to add to the cell state.
- Output Gate: Controls what part of the cell state should be output.

The operations of these gates are governed by learned weight matrices and non-linear activation functions, as detailed in the equations below.

$$f_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f h_{t-1} - 1 + \mathbf{b}_f) \quad (4)$$

$$i_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i h_{t-1} - 1 + \mathbf{b}_i) \quad (5)$$

$$\tilde{c}_t = \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c h_{t-1} - 1 + \mathbf{b}_c) \quad (6)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (7)$$

$$o_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o h_{t-1} - 1 + \mathbf{b}_o) \quad (8)$$

$$h_t = o_t \odot \tanh(c_t) \quad (9)$$

Here:

- σ is the sigmoid activation function,
- \tanh is the hyperbolic tangent activation function,
- \odot denotes element-wise multiplication,
- W and U are input and recurrent weight matrices,
- b_* are bias vectors,
- c_t is the cell state,
- c_t is the hidden state at time t .

In the context of SoC prediction, the LSTM model is trained on sequential input features extracted from battery discharge cycles—such as voltage, current, and temperature—across time. The cell state serves as the internal memory of the network, accumulating contextual information about the battery's historical behavior. This enables the model to learn complex temporal dependencies and nonlinear patterns affecting SoC, improving prediction accuracy especially when historical context is critical.

The ability of LSTM to handle sequences of varying lengths and selectively memorize relevant patterns makes it particularly suitable for real-time battery management tasks where accurate estimation of the State of Charge under dynamic load conditions is crucial.

Bi-Directional Long-Short Term Memory(Bi-LSTM) :

Bidirectional Long Short-Term Memory (Bi-LSTM) networks are an extension of the standard LSTM architecture that allow the model to have both backward and forward information about the sequence at every time step. In many sequence learning tasks, especially those involving temporal dependencies such as battery State of Charge (SoC) estimation, the future context can be just as informative as the past. Bi-LSTM addresses this by introducing two parallel LSTM layers: one processes the sequence from past to future (forward pass), and the other from future to past (backward pass). The outputs of both directions are then combined, allowing the model to capture relations more accurately.

In Bi-LSTM, for a given input sequence $X = [x_1, x_2, \dots, x_T]$, the forward LSTM computes a sequence of hidden states \vec{h}_t from $t = 1$ to T , while the backward LSTM computes hidden states \overleftarrow{h}_t from $t = T$ to 1.

These hidden states are then concatenated at each time step.

Forward LSTM

$$\vec{\mathbf{f}}_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \vec{\mathbf{h}}_{t-1} + \mathbf{b}_f) \quad (10)$$

$$\vec{\mathbf{i}}_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \vec{\mathbf{h}}_{t-1} + \mathbf{b}_i) \quad (11)$$

$$\vec{\mathbf{c}}_t = \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \vec{\mathbf{h}}_{t-1} + \mathbf{b}_c) \quad (12)$$

$$\vec{\mathbf{c}}_t = \vec{\mathbf{f}}_t \odot \vec{\mathbf{c}}_{t-1} + \vec{\mathbf{i}}_t \odot \vec{\mathbf{c}}_{t-1} \quad (13)$$

$$\vec{\mathbf{o}}_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \vec{\mathbf{h}}_{t-1} + \mathbf{b}_o) \quad (14)$$

$$\vec{\mathbf{h}}_t = \vec{\mathbf{o}}_t \odot \tanh(\vec{\mathbf{c}}_t) \quad (15)$$

Backward LSTM (similar formulation)

$$\overleftarrow{\mathbf{f}}_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \overleftarrow{\mathbf{h}}_{t+1} + \mathbf{b}_f) \quad (16)$$

$$\overleftarrow{\mathbf{i}}_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \overleftarrow{\mathbf{h}}_{t+1} + \mathbf{b}_i) \quad (17)$$

$$\overleftarrow{\mathbf{c}}_t = \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \overleftarrow{\mathbf{h}}_{t+1} + \mathbf{b}_c) \quad (18)$$

$$\overleftarrow{\mathbf{c}}_t = \overleftarrow{\mathbf{f}}_t \odot \overleftarrow{\mathbf{c}}_{t+1} + \overleftarrow{\mathbf{i}}_t \odot \overleftarrow{\mathbf{c}}_{t+1} \quad (19)$$

$$\overleftarrow{\mathbf{o}}_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \overleftarrow{\mathbf{h}}_{t+1} + \mathbf{b}_o) \quad (20)$$

$$\overleftarrow{\mathbf{h}}_t = \overleftarrow{\mathbf{o}}_t \odot \tanh(\overleftarrow{\mathbf{c}}_t) \quad (21)$$

Final Bi-LSTM output

$$\mathbf{h}_t = \begin{bmatrix} \vec{\mathbf{h}}_t \\ \overleftarrow{\mathbf{h}}_t \end{bmatrix} \quad (22)$$

In the context of SoC prediction, Bi-LSTM enables the network to consider both prior usage patterns and anticipated operating conditions of the battery. This dual-context awareness helps improve accuracy in complex scenarios—such as rapidly fluctuating load conditions or variable charging profiles—where current SoC may be affected by both past and future states. By leveraging information in both temporal directions, Bi-LSTM provides a more complete understanding of the battery’s operational sequence, making it especially suitable for time-series prediction tasks in Battery Management Systems (BMS).

Gated Rectified Unit (GRU) :

The Gated Recurrent Unit (GRU) is a simplified variant of the Long Short-Term Memory (LSTM) architecture that also belongs to the Recurrent Neural Network (RNN) family. It was proposed to address the issues of vanishing gradients and inefficient training in standard RNNs, while reducing the architectural complexity seen in LSTM. Unlike LSTM which has three gates (input, forget, output) and a separate memory cell, the GRU combines the functionalities into two gates: the update gate and the reset gate. These gates regulate the flow of information through the unit and help the network decide how much past information should be passed along to the future.

The update gate z_t determines how much of the previous state needs to be carried forward, while the reset gate r_t controls how much of the past information to forget. This enables GRU to effectively capture long-term dependencies with fewer parameters and lower computational cost than LSTM, making it an efficient alternative in scenarios like real-time battery SoC prediction.

The equations used in this method for computing hidden states are as follows:

$$\mathbf{z}_t = \sigma(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z \mathbf{h}_{t-1} + \mathbf{b}_z) \quad (\text{Update Gate}) \quad (23)$$

$$\mathbf{r}_t = \sigma(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \mathbf{h}_{t-1} + \mathbf{b}_r) \quad (\text{Reset Gate}) \quad (24)$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_h \mathbf{x}_t + \mathbf{U}_h (\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h) \quad (\text{Candidate Activation}) \quad (25)$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t \quad (\text{Final Output}) \quad (26)$$

Where:

- σ is the sigmoid activation function.
- \tanh is the hyperbolic tangent function.
- \odot denotes element-wise multiplication.
- \mathbf{z}_t : update gate vector.

- r_t : reset gate vector.
- \tilde{h}_t : candidate hidden state.
- h_t : final hidden state at time t .

In the context of battery State of Charge (SoC) prediction, GRUs serve as powerful yet lightweight sequence models. Their ability to maintain a balance between model complexity and learning capacity makes them especially useful when computational resources are limited or faster training is needed. GRUs can learn from sequential voltage, current, and temperature patterns to predict the SoC in both charge and discharge cycles, while dynamically adjusting how much past behavior influences the prediction. This is the method that performs the best in terms of SoC prediction for the chosen evaluation metrics while also taking the least amount of time to train.

State of Charge Prediction

Need for Prediction

Accurate knowledge of the State of Charge (SoC) in a battery is crucial for ensuring the safe, efficient, and reliable operation of energy storage systems, particularly in applications like electric vehicles and renewable energy integration. SoC serves as an indicator of how much usable energy remains in the battery, allowing systems to make informed decisions about charging, discharging, and load management. However, SoC cannot be directly measured using conventional sensors, as it is an internal state of the battery affected by factors such as current flow, temperature, and aging. Instead, it requires sophisticated estimation algorithms to account for dynamic variables and battery characteristics. This makes SoC estimation a critical function of modern Battery Management Systems (BMS), enabling precise control and protection of the battery through the use of mathematical models, filtering techniques, and machine learning methods. The technical challenges in SOC prediction stem from the complex variability of battery behavior with temperature, age, and current rates, alongside cell-to-cell variations and sensor limitations. Modern approaches employ increasingly advanced techniques ranging from enhanced Coulomb counting to machine learning algorithms.

Dataset Used for Training and Testing Models

The NASA Ames Prognostics Center of Excellence (PCoE) developed a comprehensive lithium-ion battery dataset to facilitate research in battery health monitoring, State of Charge (SoC) estimation, and Remaining Useful Life (RUL) prediction. This dataset comprises data from commercial 18650 lithium-ion cells(2 Ah rated capacity). The batteries were subjected to repeated charge–discharge cycles under various controlled conditions until they reached an end-of-life criterion, typically defined as a 30% capacity fade (down to 1.4 Ah)

Each battery underwent three primary operational profiles:

- Charging: Implemented using a Constant Current–Constant Voltage (CC–CV) protocol. Specifically, batteries were charged at 1.5 A until reaching 4.2 V, then maintaining 4.2 V until the charging current dropped below 20 mA.
- Discharging: Conducted at a constant current of 4 A until the battery voltage dropped to predefined thresholds, such as 2.2 V, 2.5 V, or 2.7 V, depending on the specific test conditions .
- Impedance Measurements: Periodic Electrochemical Impedance Spectroscopy (EIS) was performed to assess internal battery degradation.

The dataset encompasses a wide range of operating temperatures, including ambient (24°C), elevated (43°C), and low (4°C) conditions, to simulate real-world usage scenarios . Data acquisition was carried out using a custom-built testbed comprising of:

- Commercial 18650 Li-ion cells
- A programmable 4-channel DC electronic load and power supply
- Voltmeter, ammeter, and thermocouple sensors
- Custom EIS hardware
- Environmental chamber to impose various operational conditions
- PXI chassis based DAQ and experiment control

Each battery's data is organized into MATLAB .mat files, structured by cycles, and includes detailed time-series measurements of voltage, current, temperature, capacity, and impedance.[10]

Result Analysis

Among the tested models—Feedforward Neural Network (FFN), Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), and Gated Recurrent Unit (GRU)—the GRU model exhibited the best overall performance, achieving the lowest MSE of 0.00104, RMSE of 0.03231, and MAE of 0.01584, along with the highest R^2 score of 0.94518, indicating its strong capability in learning temporal dependencies and generalizing across test samples.

The LSTM model also performed competitively, with an MSE of 0.00136, MAE of 0.01633, RMSE of 0.03685, and an R^2 score of 0.92872. This confirms the effectiveness of its gated memory architecture in capturing time-dependent patterns in SoC sequences. The Bi-LSTM model, while slightly trailing LSTM, provided an R^2 of 0.91933, suggesting that bidirectional context marginally improves performance but may also introduce complexity that does not significantly enhance accuracy for this particular task.

On the other hand, the FFN model—which lacks inherent temporal modeling—yielded relatively higher error values with an MSE of 0.00451 and RMSE of 0.06715, and a lower R^2 score of 0.76329, underlining the importance of sequential modeling in battery SoC estimation tasks.

Overall, these results clearly demonstrate that models leveraging recurrent architectures, particularly GRU and LSTM, are more suitable for accurate SoC prediction due to their ability to retain and process temporal information across input sequences.

Solar Power Prediction

Need for Prediction

Solar photovoltaic (PV) systems, while clean and increasingly cost-effective, present challenges in integration due to their intermittent and unpredictable nature caused by weather, time of day, and seasonal variations. These fluctuations can result in grid instability, voltage and frequency disturbances, and inefficient power management, especially in standalone or distributed systems. Accurate solar power forecasting is therefore critical for effective energy scheduling, storage sizing, load balancing, and ensuring reliability in off-grid or poor-grid conditions. Forecasting not only stabilizes smart grid operations but also enhances energy management systems (EMS) by reducing reserve requirements and enabling intelligent battery charging and discharging strategies, which improve lifespan and efficiency. Traditional physical models struggle with the complex nonlinear nature of solar power generation, making machine learning approaches like LSTM, Bi-LSTM, and GRU more suitable for accurate predictions. As global PV adoption rises, integrating precise forecasting has become essential to achieving reliable, efficient, and resilient renewable energy systems.

Dataset Used for Training and Testing Models

The Solar Power Plant dataset, publicly available on Kaggle and curated by the user *pythonafroz*, offers comprehensive time-series data for analyzing and predicting solar power generation. The dataset is designed to facilitate research in renewable energy forecasting, particularly for solar photovoltaic (PV) systems, and includes detailed operational and environmental parameters from functioning solar power plants.

The dataset includes a broad range of features grouped under two main categories:

- **Solar Power Generation Parameters:**

- **DC Power (kW)** – Direct current power output from the solar panels before inversion.
- **AC Power (kW)** – Alternating current power output after conversion by the inverter.
- **Daily Yield (kWh)** – Total energy produced by the system on a given day.
- **Total Yield (kWh)** – Cumulative energy output recorded since the plant's installation.
- **Module Temperature (°C)** – Temperature of the PV modules, critical for assessing efficiency loss.
- **Ambient Temperature (°C)** – Temperature of the surrounding air at the site.
- **Irradiation (W/m²)** – Solar irradiance data received by the panels, a key factor in power generation.

- **Meteorological and Site Parameters:**

- **Temperature (°C)** – General ambient temperature observations.
- **Humidity (%)** – Atmospheric moisture content at the plant location.
- **Wind Speed (m/s)** – Wind velocity recorded on site, which may affect panel cooling and dust accumulation.
- **Power Consumption (kW)** – Real-time energy usage recorded alongside generation, useful for load balancing analysis.

Result Analysis

All three deep learning models—LSTM, GRU, and BiLSTM—achieved very similar results on the test set, with MSE values around 12,800kW², RMSE near 113kW, and R² scores close to 0.85, indicating strong but nearly identical predictive performance. GRU showed slightly better accuracy and generalization, while BiLSTM, despite its theoretical advantage of capturing bidirectional temporal patterns, performed marginally worse in this case—likely due to the relatively small dataset and limited temporal complexity. These results align with existing literature suggesting that while RNN variants often outperform classical models for short-term solar forecasting, the performance differences among them can be minimal without larger datasets or more varied input features. Given this, GRU may be preferred for its efficiency, while future improvements could focus on attention mechanisms or hybrid architectures to boost accuracy.

Conclusion

To sum up, this project presents a smart and data-driven approach to managing solar energy and battery storage systems. By combining weather-based solar forecasting with precise State of Charge (SoC) prediction using machine learning models, the system intelligently prioritizes and allocates power to critical loads. The solar generation forecasts help estimate upcoming energy availability, while the SoC predictions provide insight into the current and future battery status. These predictions are then used by a decision-making control algorithm to determine how available power should be distributed—ensuring that essential loads are prioritized and the system remains operational for as long as possible. This methodology ensures better utilization of renewable resources, minimizes energy wastage, and improves the overall reliability of standalone solar setups. It serves as a step forward toward autonomous and efficient energy management in modern renewable energy infrastructures.

Future Work

- Adding machine learning models to predict future load demand based on historical usage patterns can allow the control system to plan even more efficiently, avoiding blackouts and battery over-discharge.
- Develop the control logic with optimization algorithms (e.g., linear programming or reinforcement learning) to dynamically prioritize or defer non-essential loads based on predicted energy availability.
- Implement continuous learning where models retrain periodically on new data to adapt to seasonal changes, battery aging, or panel degradation without manual intervention.

References

1. Wan, C., Zhao, J., Song, Y., et al. "Photovoltaic and solar power forecasting for smart grid energy management." *CSEE Journal of Power and Energy Systems* 1.4 (2015): 38-46. DOI: 10.17775/CSEE-JPES.2015.00046
2. Zhang et al., MDPI 2023 (<https://www.mdpi.com/2313-0105/9/3/152>)
3. Plett, G. L., "Extended Kalman Filtering for Battery Management Systems," *Journal of Power Sources*, 2004.
4. Li-Ion Aging Dataset (<https://data.nasa.gov/dataset/li-ion-battery-aging-datasets>)