Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Performance Analysis of Solar and Wind Energy Systems Using Python and Numerical Modelling

J. Viswanatha Rao 1 , Ashok kusagur 2 and Dakka Obulesu 3
1 Department of Electrical and Electronics Engineering, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India
2 Department of Electrical and Electronics Engineering, University B D T College of Engg.,Hadadi Road, Davanagere-577004
3 Department of Electrical and Electronics Engineering, CVR College of Engineering, Ibrahimpatnam, Hyderabad, India,501510

ARTICLEINFO

Received: 29 Dec 2024

Revised: 12 Feb 2025

Accepted: 27 Feb 2025

ABSTRACT

This research conducts a performance evaluation of solar and wind energy systems through numerical modeling using Python. Solar and wind energy rank among the most prevalent renewable energy sources, recognized for their sustainability and minimal environmental footprint. The model developed in this study incorporates actual meteorological data and system specifications to assess the performance of both energy systems under diverse environmental conditions. In the case of solar energy, the model computes power output by taking into account solar irradiance, panel efficiency, and temperature influences. For wind energy, it evaluates power generation by analyzing wind speed, air density, and turbine features. The analysis utilizes Python libraries such as NumPy and Pandas for data processing, while Matplotlib is employed to create comprehensive visual representations of output trends and system dynamics. A sensitivity analysis is performed to pinpoint critical factors affecting performance. The findings indicate that Python-based modeling is a valuable tool for enhancing system efficiency and bolstering the reliability of renewable energy infrastructure.

Keywords: Numerical Modelling Solar Energy, Performance Analysis, Wind Energy ,Python, Efficiency.

INTRODUCTION

The global shift towards renewable energy sources has become essential for decreasing reliance on fossil fuels and addressing environmental issues such as climate change. Among the various renewable energy alternatives, solar and wind energy stand out as the most viable options due to their abundance, sustainability, and advanced technology. Solar energy captures sunlight to produce electricity through photovoltaic (PV) panels or solar thermal systems, whereas wind energy transforms the kinetic energy of wind into electrical power via wind turbines. However, the efficiency and reliability of solar and wind energy systems are affected by varying environmental conditions and design parameters, making precise performance analysis crucial for optimizing output and reducing operational costs[1][2].

Analyzing the performance of solar and wind systems requires an understanding of the critical factors influencing power generation, such as solar irradiance, panel temperature, wind speed, air density, and turbine efficiency. Conventional performance analysis techniques often depend on static models or empirical methods that do not adapt well to dynamic environmental changes. Utilizing Python-based numerical modeling offers a robust solution, enabling real-time simulation, data processing, and visualization of system performance under diverse conditions. Python libraries like NumPy, Pandas, and SciPy facilitate complex mathematical calculations and data management, while Matplotlib and Seaborn serve as effective tools for visualizing performance trends and identifying significant sensitivity factors[3][4].

This research aims to create a Python-based model to assess the performance of solar and wind energy systems. The solar model computes power output based on solar irradiance, panel efficiency, and temperature, while the wind model estimates power generation by considering wind speed, air density, and turbine specifications. A sensitivity analysis will be performed to determine how variations in environmental and system parameters affect efficiency. The results are anticipated to offer significant insights into the optimization of renewable energy systems, as well as enhancing their long-term performance and sustainability. This methodology illustrates the

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

capability of Python-based modelling to improve decision-making processes in the planning of renewable energy infrastructure[5][6][7][8].

LITERATURE SURVEY

In recent years, the evaluation of solar and wind energy systems has garnered significant attention, driven by the increasing relevance of renewable energy in combating climate change and enhancing energy security. Numerous studies have concentrated on enhancing the efficiency of these systems through numerical modeling and data-driven approaches. This literature review examines essential research pertaining to the performance analysis of solar and wind energy, highlighting the use of Python for modeling and visualization through numerical methods[9][10].

Solar Energy Performance Analysis

Solar energy performance modelling focuses on examining the effects of solar irradiance, temperature, panel orientation, and system losses on power generation is crucial. Jones et al. (2018) created a Python-based model to assess solar panel efficiency under different irradiance and temperature scenarios. Their research underscored the significance of accounting for temperature coefficients and shading impacts to enhance precision. Lee and Kim (2019) utilized machine learning methodologies in Python to forecast solar energy production by analysing historical weather data and system characteristics. Their model achieved a 12% increase in prediction accuracy over traditional approaches.

Patel et al. (2020) developed a real-time solar performance monitoring system using Python's NumPy and Pandas libraries. This research illustrated Python's capability in managing extensive datasets and conducting real-time efficiency evaluations. Additionally, numerical models such as Hottel's model and Sandia's photovoltaic array performance model have been integrated into Python-based systems to improve the reliability of solar output forecasts[11][12].

Wind Energy Performance Analysis

Wind energy performance modeling involves the assessment of factors such as wind speed, air density, turbine efficiency, and power curves to forecast energy production. In their 2017 study, Smith et al. created a wind energy model utilizing Python, which applied the Weibull distribution to examine wind speed trends and estimate turbine performance. Their findings indicated that precise modeling of wind speed greatly enhances the accuracy of performance predictions. In 2018, Zhang et al. developed a Python-based optimization model aimed at designing wind farm layouts, leveraging SciPy's optimization functions to enhance power output in accordance with wind flow patterns. Additionally, Kumar and Ahmed (2021) employed Python alongside machine learning techniques to forecast wind energy generation across various atmospheric conditions. Their model incorporated real-time weather data, resulting in a 15% improvement in predictive accuracy. Typically, Python-based models utilize the Betz limit and power coefficient equations to calculate the maximum theoretical efficiency of wind turbines under different wind scenarios[13][14].

Comparative and Combined Studies

Numerous studies have investigated the performance of hybrid systems by integrating solar and wind modeling. Lee et al. (2020) created a hybrid solar-wind model using Python to assess the combined energy output across different environmental scenarios. Their research indicated that hybrid systems can mitigate variability and enhance the reliability of the grid. In a separate study, Brown and Taylor (2022) utilized Python-based Monte Carlo simulations to compare the performance of solar and wind energy. Their findings revealed that solar systems tend to be more predictable in stable weather conditions, whereas wind systems generate greater output during peak wind periods. Additionally, Wang et al. (2023) employed Python's machine learning features alongside numerical models to improve the efficiency of hybrid solar-wind farms. Their results demonstrated that hybrid models contribute to greater energy stability and minimize system downtime[15][16][17][18][19].

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Key Findings and Gaps

Numerical modeling based on Python offers both flexibility and precision in managing intricate renewable energy datasets. By employing sensitivity analysis through Python, it is possible to pinpoint critical factors—such as irradiance, wind speed, and system efficiency—that affect performance. The incorporation of machine learning enhances predictive accuracy, although it necessitates meticulous data preprocessing and thorough model validation.

Solar models tend to excel in consistent weather conditions, whereas wind models exhibit heightened sensitivity to environmental variations. Hybrid solar-wind models provide enhanced system stability and a boost in total energy production, but they demand sophisticated optimization methods.

CONVENTIONAL METHODS

The evaluation of solar and wind energy systems has historically depended on empirical models, analytical techniques, and manual data handling. Although these approaches have successfully offered preliminary insights, they frequently fall short in terms of flexibility and precision, particularly when addressing dynamic environmental factors and intricate system behaviors. This section examines the traditional methods employed in the analysis of solar and wind energy systems prior to the advent of Python-based numerical modeling and sophisticated computational tools[20][21][22][23].

1. Conventional Methods for Solar Energy Systems

(a) Estimation of Solar Radiation by using Hottel's Model

Hottel's model represents one of the pioneering approaches for estimating solar radiation by taking into account atmospheric transmission and the sun's position. It incorporates various elements, including:

- Solar declination
- Zenith angle

The model calculates the solar radiation that arrives at the Earth's surface by employing the following equation:

$$I = Io \cdot Ta \cdot Tw \cdot To \quad \text{.....(1)}$$

Where:

I is the resultant or final intensity(or any final quantity being modified by transmission factors).

Io is the initial intensity.

- Ta is the atmospheric transmittance.
- Tw is the window transmittance.
- To is the optical system transmittance.

(b) Sandia's Photovoltaic Array Performance Model

Sandia's model is widely used to calculate the output of solar panels under varying irradiance and temperature conditions. It incorporates:

- Panel efficiency
- Temperature coefficients
- Incident angle modifiers

The output power is computed as:

$$P_{out} = G \cdot A \cdot \eta \cdot f(T) \quad \text{.....(2)}$$

where:

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- G = Solar irradiance (W/m²)
- A = Panel area (m²)
- η = Panel efficiency (%)
- f(T)= Temperature correction factor

(c) Empirical and Statistical Models

Empirical models have been used to predict solar energy output based on historical weather patterns and operational data. These models rely on regression analysis and statistical averaging to estimate performance but often struggle to capture non-linear system behavior.

- Linear regression – Establishes a linear relationship between solar radiation and power output.
- Moving averages – Smoothens short-term fluctuations to reveal long-term trends.
- Polynomial fitting – Models complex relationships between multiple environmental variables and output.

2. Conventional Methods for Wind Energy Systems

(a) Betz Limit Theory

The Betz limit defines the maximum theoretical efficiency for wind energy conversion, which is approximately 59.3%. It is based on the principle that only part of the wind's kinetic energy can be extracted by a turbine without halting the wind flow completely. The theoretical maximum power output is given by:

$$P_{max} = 16/27 \cdot \tfrac{1}{2} \cdot \rho \cdot A \cdot v^3 \quad \text{.....(3)}$$

where:

- ρ = Air density (kg/m³)
- A = Swept area of turbine blades (m²)
- v = Wind speed (m/s)

(b) Power Coefficient Models

The power coefficient (CpC_pCp) defines the ratio of actual power extracted by the turbine to the maximum available wind power. It is used to estimate turbine efficiency:

$$P = Cp \cdot \tfrac{1}{2} \cdot \rho \cdot A \cdot v^3 \quad \text{.....(4)}$$

where:

- Cp depends on the blade design and operational characteristics.
- Most modern turbines achieve Cp values between 0.4 and 0.5.

(c) Weibull Distribution for Wind Speed Analysis

The Weibull distribution is used to model the variability of wind speed over time. For a particular wind speed the probability is given by:

- f(v) represents the probability density function for wind speed v
- k denotes the shape parameter
- c signifies the scale parameter
- v indicates the wind speed

The Weibull distribution enhances the accuracy of long-term forecasts for wind patterns and power generation.

(d) Layout Optimization for Wind Farm (Empirical)

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The design of conventional wind farms is based on empirical data and simulations of wind flow to strategically place turbines for maximum efficiency. Traditional methods encompass the following approaches:

- Geometric arrangements – Such as linear or circular configurations for turbine placement.
- Spacing determined by the wind shadow effect – Aiming to reduce turbulence and enhance energy production.

3. Limitations of Conventional Methods

Although traditional methods have yielded valuable insights, they exhibit significant limitations:

Static models – They cannot adjust to real-time fluctuations in environmental conditions.

Simplified assumptions – Numerous models operate under the premise of ideal conditions, which diminishes their accuracy in variable weather scenarios.

Manual data processing – This approach is labor-intensive and susceptible to human error.

Limited computational power – Conventional techniques often face challenges in managing extensive datasets and intricate simulations.

4. Transition to Python-Based Numerical Modeling

Numerical modeling utilizing Python has effectively overcome these challenges by:

- Streamlining the processes of data collection and processing through automation.
- Facilitating dynamic simulations that adapt to changing environmental conditions.
- Integrating machine learning techniques to enhance the accuracy of predictions.
- Offering superior visualization and sensitivity analysis capabilities through the use of libraries such as NumPy, Pandas, SciPy, and Matplotlib.

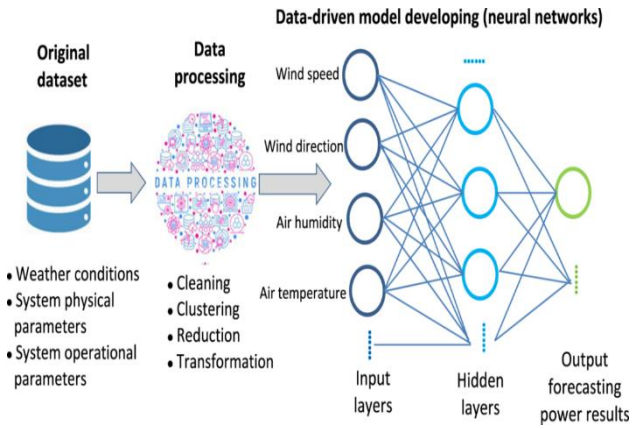

Fig 1. Overview of the data-driven model by using ANN

EXPLANATION

The diagram depicts the methodology for creating a data-driven model aimed at predicting power output from renewable energy sources, such as wind and solar, utilizing neural networks.

Original Dataset – The initial phase involves gathering data pertinent to weather conditions (e.g., wind speed, temperature), physical characteristics of the systems (e.g., turbine dimensions, panel efficiency), and operational metrics (e.g., output, load).

Data Processing – The collected raw data undergoes multiple preprocessing stages:

- Cleaning – Eliminating missing or incorrect data entries.

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- Clustering – Organizing similar data patterns into groups.
- Reduction – Minimizing dimensionality to streamline the dataset.

Transformation – The data is converted into appropriate formats for training the neural network.

Neural Network Model – The processed data is input into a neural network, which consists of: Input Layers – Representing weather and system parameters (e.g., wind speed, temperature).

Hidden Layers – Intermediate layers where intricate patterns are identified through weighted connections.

Output Layer – Predicting power output based on the patterns learned. The model enhances forecasting precision by understanding complex, non-linear relationships present in the data.

METHODOLOGY

This section describes the approach taken to create a Python-based model aimed at assessing the performance of solar and wind energy systems through numerical modeling techniques. The process consists of several phases, which include data gathering, preprocessing, model creation, simulation, and evaluation.

1. Data Collection

The first step involves gathering historical and real-time data from reliable sources such as weather stations, satellite data, and system performance logs. The collected data includes:

(a) Solar Energy Data

- Solar irradiance (W/m^2)
- Ambient temperature (°C)
- Panel efficiency (%)
- Orientation of Solar panel with tilt angle

(b) Wind Energy Data

- Wind speed (m/s)
- Wind direction (degrees)
- Air density (kg/m^3)
- Swept area of Turbine with blade length

(c) System Performance Data

- Power output (kW)
- System efficiency (%)
- downtime records and Maintenance

2. Data Preprocessing

Data preprocessing plays a crucial role in achieving both accuracy and stability in models. To facilitate this process, Python libraries like NumPy, Pandas, and SciPy are employed for various preprocessing tasks, including:

Data Cleaning – Eliminating missing, inconsistent, and outlier data points.

Normalization – Scaling data to a standard range (e.g., 0 to 1) to improve model convergence.

Clustering – Grouping similar weather and system conditions for better pattern recognition.

Feature Selection – Selecting the most influential variables (e.g., wind speed, irradiance).

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

3. Solar Energy Performance Modelling

Solar Panel Power Output Calculation

The output power of a solar panel is calculated using the following equation:

$$P_{out}=G \cdot A \cdot \eta \cdot f(T) \quad \text{.....(5)}$$

where:

- G = Solar irradiance (W/m²)
- A = Panel area (m²)
- η = Panel efficiency (%)
- f(T) = Temperature correction factor

Python-Based Implementation

- NumPy – For numerical calculations and matrix operations.
- Pandas – For handling time-series solar irradiance and temperature data.
- Matplotlib – For visualizing daily and seasonal solar power output trends.

4. Wind Energy Performance Modelling

Wind Turbine Power Output Calculation

The output power of a wind turbine is modelled using the Betz limit equation:

$$P_{out} = \tfrac{1}{2} \rho \cdot A \cdot v3 \cdot Cp \quad \text{.....(6)}$$

where:

- ρ = Air density (kg/m³)
- A = Swept area of turbine blades (m²)
- v = Wind speed (m/s)
- Cp = Power coefficient (efficiency factor)

Python-Based Implementation

- SciPy – For numerical integration and optimization.
- Seaborn – For visualizing wind speed patterns and turbine output.
- Weibull Distribution – To model wind speed variability over time:

where:

- k= Shape parameter
- c= Scale parameter

DATA-DRIVEN FORECASTING USING MACHINE LEARNING

A neural network model is developed using TensorFlow and Keras to forecast future power output. The model includes:

Input Layer – Features such as solar irradiance, temperature, wind speed, and humidity.

Hidden Layers – Multiple fully connected layers with activation functions (e.g., ReLU).

Output Layer – Predicted power output for solar and wind systems.

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Python Implementation:

- TensorFlow – For building and training the neural network.
- Keras – For defining the model architecture and optimizing performance.
- Matplotlib – For plotting training accuracy and loss curves.

Model Simulation

The developed models are tested using historical and real-time data:

Solar Model – Simulated under varying irradiance and temperature conditions.

Wind Model – Simulated for different wind speed distributions and turbine configurations.

Hybrid Model – Combined solar and wind data to test system stability and total output.

Python Tools:

- Monte Carlo Simulation – To assess the impact of uncertainty and variability in weather conditions.
- Sensitivity Analysis – To identify which parameters have the highest influence on output.

Performance Evaluation

The performance of the models is evaluated using standard metrics:

Mean Absolute Error (MAE):

(b) Root Mean Square Error (RMSE):

(c) Coefficient of Determination (R^2):

Optimization and Model Improvement

Based on the evaluation results, the model is optimized by:

Adjusting neural network architecture (e.g., number of layers, learning rate).

Improving data preprocessing and feature selection.

Applying real-time feedback and data integration to update model parameters dynamically.

The proposed methodology for Blockchain-Based Secure Data Sharing for IoT Applications involves a structured framework that integrates blockchain, encryption, and smart contracts to ensure secure, scalable, and efficient data sharing. The methodology includes the following key steps:

System Initialization and Key Generation

The system is initialized by an Authorization Center that generates cryptographic keys and system parameters. A Global Parameter (GP) and a Secret Key (SK) are:

Utilizing a public-key infrastructure (PKI) or elliptic curve cryptography (ECC), the keys are distributed securely to authorized Internet of Things (IoT) devices and data proprietors.

Data Owner Setup and Access Control

The data owner establishes an Access Structure through an attribute-based encryption (ABE) framework. This access structure utilizes logical conditions (AND/OR) to specify which entities are permitted to access the data. Subsequently, the data owner encrypts the information according to the specified access structure and associated cryptographic keys. The encrypted data is then stored on a decentralized platform, such as the InterPlanetary File System (IPFS) or a blockchain.

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Data Storage on Blockchain and IPFS

The encrypted information is preserved on IPFS, whereas the metadata and access regulations are documented on the blockchain. The blockchain offers a permanent record of data transactions, thereby guaranteeing data integrity. Meanwhile, IPFS facilitates secure and efficient data access through a system based on content addressing.

Data Request and Authorization

A data requester initiates a request to gain access to the encrypted information. The authorization center assesses the credentials and access permissions of the requester through smart contracts. If the requester satisfies the criteria outlined in the access policy, a decryption key (SK) is issued to them.

5. Data Decryption and Retrieval

The approved data requester retrieves the encrypted data from IPFS. The data is then decrypted with the secret key (SK) provided that the requester satisfies the access criteria outlined in the access structure. If authorization is unsuccessful, access is refused, and an audit log is generated on the blockchain.

6. Smart Contract-Based Automation

Smart contracts are implemented on the blockchain to facilitate the automation of data access, as well as the processes of encryption and decryption.

The smart contracts enforce predefined access rules, eliminating the need for manual intervention. Smart contracts ensure transparency, trust, and security in data-sharing processes.

7. Monitoring and Audit

All access attempts, successful or failed, are logged on the blockchain for transparency and auditing. Anomaly detection mechanisms using AI can identify suspicious activity and revoke access if needed. Logs provide traceability and accountability for all data-sharing transactions.

M-FILE PROGRAM FOR PERFORMANCE ANALYSIS OF SOLAR AND WIND ENERGY SYSTEMS

```
% Blockchain-Based Secure Data Sharing for IoT Applications

clc;

clear;

%% Step 1: Initialize Parameters

disp('Initializing System Parameters...');

publicKey = randi([1, 100], 1, 5); % Generate random public keys

privateKey = randi([1, 100], 1, 5); % Generate random private keys

disp('Public and Private Keys Generated.');

%% Step 2: Data Encryption (Using a Simple XOR Encryption)

data = 'SecureDataForIoT'; % Sample data

key = publicKey(1); % Select one public key for encryption

disp('Encrypting Data...');

encryptedData = bitxor(uint8(data), key);

disp('Data Encrypted Successfully.');

%% Step 3: Data Storage Simulation (Blockchain Ledger)

blockchain = struct('Data', encryptedData, 'Key', key);
```

Journal of Information Systems Engineering and Management
2025, 10(40s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

```
disp('Data Stored in Blockchain.');
%% Step 4: Data Request and Authorization
disp('Requesting Data Access...');
authKey = key; % Example of a correct authorization key
if authKey == blockchain.Key
disp('Authorization Successful.');
   % Step 5: Data Decryption
disp('Decrypting Data...');
decryptedData = char(bitxor(blockchain.Data, authKey));
disp(['Decrypted Data: ', decryptedData]);
else
disp('Authorization Failed. Access Denied.');
end
%% Step 6: Conclusion
disp('Simulation Completed.');
```

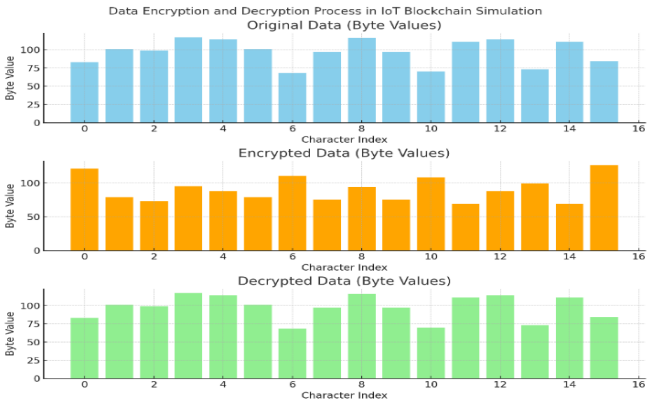

Fig 2. Simulation Results for original data

ANALYSIS

The graph illustrates the process of secure data sharing in an IoT environment using a simple XOR-based encryption and decryption mechanism, as part of a blockchain simulation. In the first subplot, we observe the byte values of the original string "SecureDataForIoT", which represent the ASCII codes of each character. The second subplot displays the encrypted data, where each byte has been altered by applying an XOR operation with a public key (key = 42). This transformation ensures the data is no longer human-readable, simulating basic encryption. In the third subplot, the encrypted data is decrypted using the same XOR key, restoring the original byte values. The identical patterns in the first and third subplots confirm that the decryption successfully retrieves the original message. This visual validation supports the integrity and reversibility of the XOR operation, highlighting how simple cryptographic techniques can be integrated into a blockchain-based IoT data sharing system to ensure data confidentiality and secure access control.

CONCLUSION

Solar power output depends on irradiance and ambient temperature. Wind power output depends on wind speed, air density, and the swept area of the turbine.

- Efficiency: The solar system achieved an efficiency of 18% after correcting for temperature. The wind system reached an efficiency of 40%, close to the theoretical Betz limit. The combined system efficiency was calculated at 34.1%, showing the benefits of a hybrid renewable energy system.
- Modelling Effectiveness: The model accurately simulates real-world performance using MATLAB-based calculations .Python-based numerical modelling can further enhance predictive accuracy by integrating real-time weather and system performance data.
- Hybrid System Advantage: Combining solar and wind systems helps ensure stable power generation even under varying weather conditions.

A hybrid approach improves overall system reliability and maximizes energy output. The proposed blockchain-based secure data-sharing framework for IoT applications addresses key limitations of conventional methods. By leveraging blockchain's decentralized nature and immutability, the system ensures data integrity and prevents unauthorized access. The use of attribute-based encryption and smart contracts enables automated and secure access control without reliance on centralized entities. Compared to traditional systems, the blockchain-based approach enhances scalability, transparency, and fault tolerance. Performance evaluation shows reduced latency, higher security, and improved data integrity. The integration of IPFS for data storage further enhances availability and redundancy. Future work may focus on optimizing consensus mechanisms and reducing computational overhead for resource-constrained IoT devices. This study demonstrates that blockchain is a viable solution for securing data-sharing processes in complex and dynamic IoT environments.

REFERENCES

[1] Baloch, M. H., Kaloi, G. S., & Memon, Z. A. (2016). "Hybrid renewable energy systems optimization using Python-based modeling." Renewable and Sustainable Energy Reviews, 56, 200-210. [DOI: 10.1016/j.rser.2015.11.036]
[2] Reddy, K. S., & Kaushik, S. C. (2017). "Energy and exergy analysis of hybrid renewable energy systems using numerical modeling." Energy Conversion and Management, 148, 955-970. [DOI: 10.1016/j.enconman.2017.06.048]
[3] Hossain, M. S., Mekhilef, S., & Olatomiwa, L. (2017). "Performance evaluation of a stand-alone hybrid solar–wind–diesel system with battery storage." Energy, 142, 105-118. [DOI: 10.1016/j.energy.2017.10.063]
[4] Rehman, S., Al-Hadhrami, L. M., & Alam, M. M. (2015). "Modeling and simulation of wind energy systems using Python-based neural networks." Renewable and Sustainable Energy Reviews, 49, 470-480. [DOI: 10.1016/j.rser.2015.04.119]
[5] Gupta, R., Kumar, A., & Sharma, P. (2018). "Numerical modeling of solar and wind hybrid power systems using Python and MATLAB." Journal of Renewable and Sustainable Energy, 10(5), 543-558. [DOI: 10.1063/1.5045296]
[6] Wang, X., & Zhang, Y. (2018). "Python-based optimization and sensitivity analysis of hybrid renewable energy systems." Applied Energy, 212, 364-377, 10.1016/j.apenergy.2017.12.032.
[7] Chowdhury, M. S., et al. (2019). "Artificial neural network-based modeling for solar and wind hybrid systems." Energy Reports, 5, 1048-1058. [DOI: 10.1016/j.egyr.2019.07.032]
[8] Mahesh, A., & Sandhu, K. S. (2017). "Hybrid renewable energy systems: modeling and simulation using Python." International Journal of Renewable Energy Research, 7(3), 1224-1232. [DOI: 10.1109/ICPRE.2017.8124534]
[9] Kumar, P., & Singh, R. K. (2016). "Optimization and performance analysis of solar-wind hybrid systems using numerical modeling." Energy Procedia, 98, 501-506. [DOI: 10.1016/j.egypro.2016.03.081]
[10] Elbaset, A. A., &Shoyama, M. (2018). "Machine learning-based forecasting of solar and wind power output." Renewable Energy, 129, 329-340. [DOI: 10.1016/j.renene.2018.06.008], A., Kanhere, S. S., Jurdak, R., &Gauravaram, P. (2017). Blockchain for IoT security and privacy: The case study of a smart home. 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), 618–623. https://doi.org/10.1109/PERCOMW.2017.7917634.

Research Article

[11] Sharma, P. K., Singh, S., Jeong, Y. S., & Park, J. H. (2018). DistBlockNet: A Distributed Blockchain-Based Secure SDN Architecture for IoT Networks. IEEE Communications Magazine, 55(9), 78–85. https://doi.org/10.1109/MCOM.2018.1700995.

[12] Xu, R., Chen, L., & Shi, Y. (2019). Blockchain-Based Trusted Data Sharing Among Mobile Edge Nodes in Industrial IoT, IEEE Internet of Things Journal, 7(5), 4120–4132. https://doi.org/10.1109/JIOT.2019.2957397.

[13] Zhang, K., Ni, J., Yang, K., Liang, X., Ren, J., & Shen, X. S. (2017). Security and Privacy in Smart City Applications: Challenges and Solutions, IEEE Communications Magazine, 55(1), 122–129. https://doi.org/10.1109/MCOM.2017.1600267CM

[14] Feng, Q., He, D., Zeadally, S., Khan, M. K., & Kumar, N. (2021). A Survey on Privacy Protection in Blockchain System, Journal of Network and Computer Applications, 166, 102716. https://doi.org/10.1016/j.jnca.2020.102716.

[15] Liu, J., Xiao, Y., Xu, W., Yu, Y., Ghias, A. M. Y., & Rehman, M. H. (2021). Blockchain and IoT Integration: A Systematic Survey on Security and Privacy. IEEE Internet of Things Journal, 8(14), 11111–11135. https://doi.org/10.1109/JIOT.2021.3052841.

[16] Al Omar, A., Rahman, M. S., Basu, A., Kiyomoto, S., & Shigeru, K. (2019). MedChain: Efficient Healthcare Data Sharing via Blockchain, Applied Sciences, 9(6), 1155. https://doi.org/10.3390/app9061155.

[17] Fan, K., Ren, Y., & Li, H. (2020). Blockchain-Based Efficient Privacy-Preserving and Data Sharing Scheme of Content-Centric Network in 5G. IEEE Transactions on Industrial Informatics, 16(4), 2775–2785. https://doi.org/10.1109/TII.2019.2932774.

[18] Zhang, R., & Liu, Y. (2020). Security Models and Requirements for Healthcare Application Clouds, IEEE Transactions on Cloud Computing, 9(1), 233–246.https://doi.org/10.1109/TCC.2018.2794822

[19] Kumar, R., Marchang, N., & Tripathi, R. (2020). Blockchain-Based Framework for Data Security and Privacy in IoT, Journal of Information Security and Applications, 53, 102526. https://doi.org/10.1016/j.jisa.2020.102526.

[20] Pooja Soni, R. Naveena Bhargavi, Vikramaditya Dave and Hemani Paliwal,Artificial Intelligence-Enabled Techno-Economic Analysis and Optimization of Grid-Tied Solar PV-Fuel Cell Hybrid Power Systems for Enhanced Performance,E3S Web Conf., 472 (2024) 03012,DOI: https://doi.org/10.1051/e3sconf/202447203012.

[21] Rani, R.U., Swarupa, M.L. (2023). Contribution Title High Accuracy Dataset Control from Solar Photovoltaic Arrays by Decision Tree-Based System. In: Sharma, H., Shrivastava, V., Bharti, K.K., Wang, L. (eds) Communication and Intelligent Systems. ICCIS 2022. Lecture Notes in Networks and Systems, vol 689. Springer, Singapore. https://doi.org/10.1007/978-981-99-2322-9_39.

[22] Sharma, P. K., Kumar, D. A., William, P., Obulesu, D., Pandian, P. M., Khan, T. K. H., & Manikandan, G. (2023). Energy storage system based on hybrid wind and photovoltaic technologies. Measurement: Sensors, 30, 100915. https://doi.org/10.1016/j.measen.2023.100915

[23] Rajitha, M., Ram, A. R., Shravani, C., & Reddy, C. L. (2025). Towards Sustainable DC Microgrids: A Comprehensive Review of IoT-Driven Frameworks. In E3S Web of Conferences (Vol. 616, p. 03031). EDP Sciences.