Optimizing Power Efficiency in Cloud Data Centers: An Integrated Machine Learning and Dynamic Threshold Approach

Ganesha Eshwar, Sai Keertana Padmanabham
Department of Computer Science, California State University, Dominguez Hills, CA, USA
E-mail: geshwar1@toromail.csudh.edu, spadmanabham1@toromail.csudh.edu

***Abstract*—With the growing reliance on cloud computing, optimizing the electrical energy efficiency of cloud data centers has become increasingly vital. This project presents an energy-efficient hybrid (EEH) framework to enhance power usage efficiency in cloud data centers. Distinct from existing methodologies that rely solely on either request scheduling or server consolidation, the EEH framework integrates both. It starts by sorting customer requests based on their time and power requirements, followed by a scheduling algorithm that takes power consumption into account. A consolidation algorithm then identifies underloaded and overloaded servers, determining the virtual machines to be migrated and the destination servers. Additionally, a migration algorithm facilitates the transfer of these virtual machines. The framework's efficacy is demonstrated through a Python implementation, using a synthetic dataset to simulate a cloud data center environment. The performance is evaluated using a linear regression model, with dynamic threshold adjustments based on factors like peak hours and server load. This approach is validated through various metrics such as power usage effectiveness (PUE) and data center energy productivity (DCEP). The results, visualized through comparative graphs, indicate the framework's potential in reducing power consumption, thereby contributing to more sustainable cloud computing operations.**

Keywords—Green cloud computing; energy efficiency; green computing; linear regression; power consumption; data center management; dynamic thresholding; virtual machine migration; server consolidation; workload scheduling.

I. INTRODUCTION

The digital transformation era has resulted in an unprecedented boom in data center deployments due to the surge in cloud computing services. Although the foundation for recent digital services, these data centers also provide substantial electrical energy consumption concerns. Energy-efficient solutions are desperately needed in cloud data centers due to the negative effects this energy use is having on the environment and the economy [1][2]. The latest advances in cloud computing prioritize more environmentally friendly and sustainably operated processes in addition to improving performance and scalability. One important strategy that aims to minimize data center energy use and carbon footprint is the idea of "Green Cloud Computing." This is helpful for service providers financially as well as for the environment [3].

The paper introduces a new architecture for Energy-Efficient Hybrids (EEHs) that optimize power consumption in cloud data centers. The EEH framework incorporates both request scheduling and server consolidation, in contrast to conventional methods that only use one of these. It begins by intelligently classifying client requests according to their power and time needs. A scheduling algorithm then takes power consumption into account while making decisions. Furthermore, a consolidation algorithm is used to determine which servers are overloaded or underloaded, allowing virtual machines to be strategically moved to optimize server loads [4][5].

In order to assess the effectiveness of the EEH framework, a machine learning model—specifically, linear regression—is used. Based on a number of variables, including resource type, time required, and server load, this model forecasts power consumption. The dynamic thresholding aspect of the model improves its practical application in real-world circumstances by adjusting forecasts depending on real-time data, such as peak hours and server load. This study showcases the potential of the EEH framework in decreasing power usage and supporting sustainable cloud computing operations by applying the model through rigorous testing and visualizations [6][7].

This approach addresses the energy issues faced by cloud data centers, which advances both the technical domain of cloud computing and the more general objectives of environmental sustainability. The EEH framework, which offers a balanced approach between technological innovation and ecological responsibility, prepares the path for future research and development in energy-efficient cloud computing.

This paper is structured as follows: Section II, "Related Work," reviews prior research and its limitations that inform our study. Section III, "Proposed Approach," links these insights to our novel methods and algorithms. Section IV presents the "Results" of our implementation, and Section V, "Conclusion and Future Work" offers a retrospective summary and implications of our findings. The paper concludes with the "References" section.

II. RELATED WORK

Previous research on green cloud computing focused on scheduling tasks effectively, consolidating virtual machines, and allocating resources dynamically in order to maximize energy utilization. A promising development in operational efficiency has been demonstrated by machine learning algorithms for predictive energy management. But comprehensive answers to practical problems are rarely found in the literature that currently exists. By fusing server consolidation and request scheduling, as well as augmenting these with a machine learning-based method for power consumption prediction, the suggested Energy-Efficient Hybrid (EEH) system seeks to overcome these drawbacks. This strategy seeks to advance cloud data center operations that are more sustainable and efficient.

A. Motivation

The rapid growth of cloud computing, driven by digital transformation, has resulted in a notable rise in the quantity and size of data centers across the globe. Despite being the foundation of contemporary digital services, these centers now consume a substantial amount of electricity [1]. In addition to posing questions regarding environmental sustainability, this increase in energy use places an extensive cost on service providers [2]. Green computing techniques are becoming increasingly important as people become increasingly mindful of how energy usage affects climate change [3].

As a result, the search for energy-efficient cloud data center solutions is a vital aspect of both technological growth and social responsibility, not only an environmental or financial issue [4]. The carbon footprint of the data centers, which increases greenhouse gas emissions worldwide, emphasizes this urgency even more [5]. As a result, the idea of "Green Cloud Computing" has surfaced, supporting the incorporation of environmentally friendly methods into the planning, execution, and retirement of cloud infrastructure [6]. By minimizing energy use and lowering data centers' carbon footprint, this strategy seeks to balance scientific advancement with environmental protection.

Our objective is to reduce energy usage and emissions for environmental reasons, cut operating expenses for financial reasons, and take on the technological challenge of optimizing cloud infrastructure are the three main drivers for our initiative [9]. This work tries to provide a significant answer to these urgent problems by creating an Energy-Efficient Hybrid (EEH) framework that intelligently manages power utilization in cloud data centers [10]. Increasing cloud computing energy efficiency through a comprehensive and creative approach is illustrated by the framework's combination of request scheduling, server consolidation, and machine learning-based power consumption prediction [11].

B. Literature Review

The essential need of satisfying the energy demands of cloud data centers that are expanding at an accelerated rate has drawn a great deal of attention to the field of green cloud computing. The important areas that have shaped the current approaches to improving cloud computing's energy efficiency are examined in this survey of the literature.

Dynamic Resource Allocation: Research on dynamic resource allocation focuses on optimizing computational resources to reduce idle time and unnecessary energy usage. This strategy is particularly effective in cloud systems, where unpredictable workload patterns can lead to significant energy savings by dynamically modifying resources in response to current demand.

Virtual Machine Consolidation: The study demonstrates that strategically placing virtual machines (VMs) among physical servers can increase energy efficiency by maximizing usage while consuming the least amount of energy, as demonstrated by various methods for virtual machine consolidation [2].

Effective Workload Scheduling: In cloud data centers, the effective scheduling of workloads in cloud data centers significantly impacts energy management. Algorithms developed in this field ensure equitable computing load distribution among servers, preventing overwork and underutilization, thus reducing total energy usage in cloud infrastructures [1].

Green Cloud Computing Frameworks: Present studies are focusing on creating comprehensive frameworks that include a range of energy-saving techniques. Through the integration of strategies like workload scheduling, virtual machine consolidation, and resource allocation, these frameworks seek to offer a complete approach to energy efficiency. [4] emphasized their importance in developing sustainable cloud computing.

Machine Learning for Energy Prediction: With the development of machine learning, new avenues for anticipating and controlling data center energy consumption have become possible. Machine learning models can be used to predict energy requirements with high accuracy. [5]. This allows for proactive resource management modifications.

C. Limitations

The project aims to solve various constraints in present research on green cloud computing, despite significant improvements in the field.

Narrow Focus on Individual methods: A large portion of the literature now in publication, including research [2], focuses on specific methods like dynamic resource allocation or virtual machine consolidation. Although these approaches work well on their own, they frequently fail to take into account how linked data center operations are, which can result in solutions that prioritize one element over others.

Insufficient Handling of Dynamic Workloads: It is often the case that existing frameworks and algorithms are unable to dynamically adjust to evolving server conditions and workloads. Resource allocation has advanced [3], but these methods are not always flexible enough to react quickly to changes in workload intensity, which leads to inefficiencies and higher energy use.

Deal-offs Performance and Energy Efficiency: Finding a way to balance energy savings with performance indicators like execution time and availability is a typical problem in energy-efficient cloud computing. In the paper [4] they have pointed out that attempts to cut back on energy use may unintentionally affect service quality, therefore a more balanced strategy is required.

Underutilization of Predictive Analytics: Many current frameworks fall short of realizing the full promise of predictive analytics, particularly machine learning, in optimizing data center operations. The application of data center software enhancements for energy efficiency [4], [5], yet there is still a lack of wider integration of predictive models in operational decision-making.

Limited Focus Made to Wider Environmental Effects: Although cutting energy use is a top priority, the wider environmental effects—such as resource use and carbon emissions—are frequently neglected. While studies say to address this by bringing in a broader ecological perspective, more thorough attention is still needed in this area [6].

Our project's proposed Energy-Efficient Hybrid (EEH) framework is made to get around these restrictions. The EEH framework provides a more comprehensive, flexible, and ecologically conscious response to the problems associated with energy efficiency in cloud computing by combining request scheduling with server consolidation and augmenting these procedures with a machine learning-based method for power consumption prediction.

III. PROPOSED APPROACH

Our proposed solution, the Energy-Efficient Hybrid (EEH) architecture, is a thorough method intended to greatly improve cloud computing environments' energy efficiency. This solution combines a number of essential elements in a unique manner, each designed to handle a certain facet of energy consumption in cloud data centers.

A. Key Components of Proposed System

Following are the key components of the EEH Framework model:

User Request Interface: This interface serves as the initial point of interaction for users. It receives and processes cloud service requests, which may include specifications like computational resource needs, expected duration, and any special power constraints. It's essential for accurately capturing user needs and translating them into quantifiable parameters that can be processed by subsequent components of the framework.

Sorting Algorithm: This algorithm classifies requests according to their time and power requirements as soon as they are received. This classification is essential for determining which jobs require more energy to complete and should be handled in a priority manner. The sorting algorithm is essential to controlling the system's power efficiency since it makes sure that the most demanding jobs are completed in a way that maximizes energy consumption all around.

Scheduling Algorithm: This algorithm assigns the sorted requests to the relevant servers. It's not only about being available; it's also about distributing the load and using the least amount of energy possible when matching requests to servers. In order to prevent any one server from being overworked or underutilized and to improve the overall energy efficiency of the data center, effective scheduling is essential.

Server Consolidation: The Algorithm continuously evaluates the load on servers to determine which ones are overburdened or underloaded. Following that, it consolidates workloads to make sure servers are used as efficiently as possible—possibly by VM migration. Energy can be saved by lowering the number of active servers at any given moment through server consolidation. Additionally, it reduces the operating expenses linked to maintaining idle servers.

Machine Learning Using Linear Regression: The machine learning module forecasts the power consumption of servers using a linear regression model by taking into account a number of different inputs, including the kind and quantity of active requests, the condition of the server, and past patterns of power usage. For proactive energy management, this predictive ability is essential. It enables the system to schedule workloads and allocate resources intelligently depending on anticipated patterns of energy use.

B. System Architecture

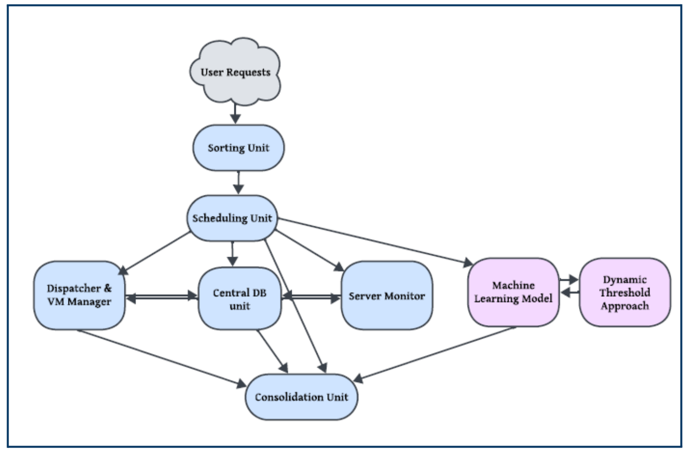

Figure 1 Proposed Architecture

User Requests: Service requests are the first step in the cloud data center's processing chain, and they are how users engage with the system.
Sorting Unit: Type and priority are used to categorize incoming service requests. Before the requests are scheduled for processing, this unit makes sure they are arranged.
Scheduling Unit: After receiving the sorted requests, this unit allocates them to the proper data center resources so they can be carried out.
Dispatcher & VM Manager: This part is in charge of allocating tasks to the virtual machines (VMs) while maintaining track on how well they're using their resources.
Server Monitor: Works in combination with the Dispatcher & VM Manager, continuously monitoring the health and performance of servers and supplying vital information for changes to operations.
Central Database Unit: Serves as the primary collecting area for all operational data, including logs of resource allocation, energy usage, and virtual machine statuses.
Server Cluster: The context suggests that the Server Monitor is in charge of a cluster of servers (Server Cluster), each of which is hosting many virtual machines (VMs), even though this isn't stated directly in the flow.
Consolidation Unit: To improve virtual machine (VM) allocation and balance the load among servers in an effort to lower energy consumption, this unit uses real-time data from the Server Monitor.
Machine Learning Model: It applies models such as linear regression to forecast future power consumption based on operational data, both historical and current, from the Central Database Unit.

Algorithm: Synthetic Time-Series Data Generation and Splitting for Power Consumption Analysis

Input:
num_samples: Integer, total number of data points to generate.
start_date: String, starting date for the time-series data.

Output:
train_data: DataFrame, training dataset containing 80% of the generated samples.

test_data: DataFrame, testing dataset containing the remaining 20% of the samples.

Initialization:
Set the random seed for reproducibility to a fixed value.
Initialize the number of samples (num_samples) and the starting date (start_date) for the time-series.

Procedure:

- Generate a date range from the specified start_date to the calculated end_date with daily frequency.
- Simulate the CPU usage (cpu_usage) as a uniform distribution between predefined lower and upper bounds.
- Generate the number of active servers (active_servers) as random integers within a specified range.
- Create a time-of-day pattern (time_of_day) using a sine function to simulate a natural daily cycle.
- Calculate the power consumption (power_consumption) as a linear combination of cpu_usage, active_servers, and time_of_day, with an added Gaussian noise for variability.
- Aggregate the date_range, cpu_usage, active_servers, time_of_day, and power_consumption into a pandas DataFrame named data.
- Split data into training (train_data) and testing (test_data) sets based on a predetermined ratio.

End Algorithm

IV. RESULTS AND DISCUSSION

The implementation of a linear regression model using the scikit-learn library has yielded promising results. The model was trained on the synthetic dataset to predict power consumption based on features that are representative of a cloud data center's operational parameters.

A. Model Evaluation

The model's performance was rigorously evaluated on a separate test set using standard metrics. The Mean Squared Error (MSE) and R-squared (R^2) values were computed to quantify the model's accuracy and its ability to explain the variance in power consumption. The training Root Mean Squared Error (RMSE) was calculated at 4.940876335571659, which provides a measure of the average error magnitude in the same units as the power consumption.

The test results showed an MSE of 25.104161760187225, suggesting a satisfactory level of precision in the predictions. Furthermore, an R-squared value of 0.9163845109242488 on the test set indicates a strong correlation between the predicted and actual power consumption values, with the model explaining over 91% of the variance.

B. Dynamic Threshold Implementation

Including dynamic thresholds according to the "Time of Day" was an essential component. By reflecting peak and off-peak hours in the model's thresholds, the sensitivity of the model to changes in operations was improved. Given that daily and seasonal swings affect power consumption patterns in cloud data centers, this adaptability is extremely helpful.

C. Time Series Analysis

The time series analysis provided deeper insights into the model's predictive capabilities over time (Figure 1, Figure 2 and Figure 3). The visualizations of actual vs. predicted power consumption show a close alignment, with the predicted values mirroring the actual consumption trends.

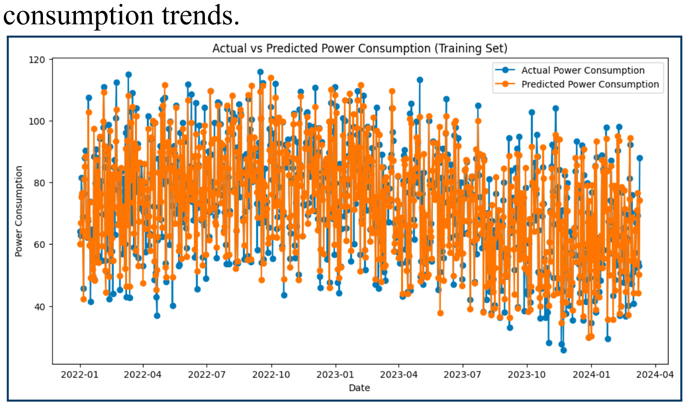

Figure 2 Actual vs Predicted Power Consumption (Training Set)

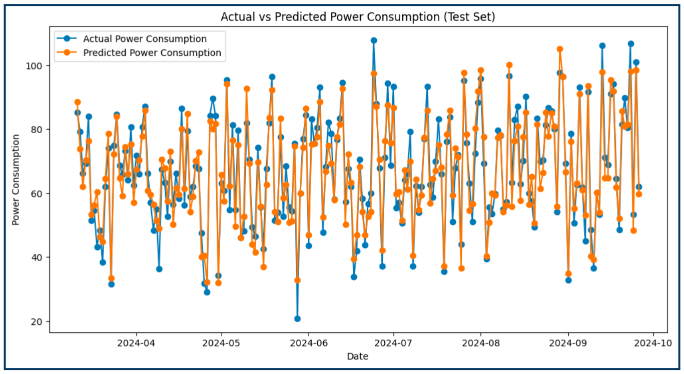

Figure 3 Actual vs Predicted Power Consumption (Test Set)

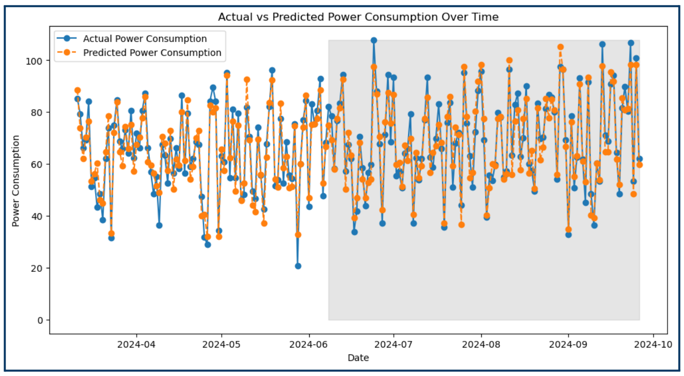

Figure 4 Actual vs Predicted Power Consumption Over Time

The analysis of peak hours and the identification of anomalies were particularly insightful. The model effectively highlighted periods of increased power usage, which are crucial for energy management and planning. Anomalies detected between the predicted and actual values could suggest areas for further model refinement or the presence of unaccounted factors influencing power consumption.

D. Discussion

The results show how well the model can predict power consumption trends and offer useful, actionable insights. Proactive energy management techniques in cloud data centers are made possible in large part by this predictive power. A crucial prerequisite for real-world applications is the model's flexibility in responding to temporal fluctuations, which is further highlighted by the dynamic thresholding.

It is essential to understand that even though the model works well with synthetic data, real-world validation is required to verify its usefulness. Furthermore, investigating more complex models and adjusting the model's hyperparameters might result in even greater performance gains.

E. Cloud Data Center Management Implications

For cloud data center management, the capacity to accurately predict power usage has important ramifications. Better planning for energy use, more effective resource allocation, and maybe lower operating costs are all possible outcomes. Additionally, the knowledge gained from this model might help develop plans to lessen the negative effects of data center operations on the environment.

V. CONCLUSION AND FUTURE WORK

The Energy-Efficient Hybrid (EEH) framework provided in this paper is intended to improve cloud data center sustainability by utilizing intelligent power usage prediction. The application of a linear regression model, which revealed a great ability to estimate energy usage accurately, as proven by the low MSE and high R^2 values acquired during testing, demonstrated the effectiveness of the framework.

The time series analysis's findings confirmed the model's resilience even more, especially in terms of its ability to represent the periodic patterns of power consumption linked to peak and off-peak times. The model's capacity to adjust to real-time operational situations was highlighted by the dynamic thresholding technique, which distinguishes it from static models that are unable to provide the flexibility required in dynamic cloud environments.

The results have important ramifications and open up new avenues for more economical and energy-efficient cloud data center operations. In line with the rising demand for green computing solutions, the use of such predictive models can result in more strategically allocated resources, cheaper operating costs, and a smaller environmental effect.

Future Work: This research aims to optimize cloud computing environments by focusing on real-world data validation, advanced predictive models, hyperparameter optimization, integration with renewable energy sources, comprehensive energy management systems, scalability and adaptability, and educational and regulatory contributions. The next phase will involve validating the model with real-world data to ensure its applicability and accuracy in operational environments. The ultimate goal is to integrate the EEH framework into a broader energy management system that includes real-time monitoring and automated control mechanisms. The research also aims to contribute to educational efforts and shape regulations around energy efficiency in cloud computing.

APPENDIX

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split

# Set random seed for reproducibility
np.random.seed(42)

# Generate synthetic time-series data
num_samples = 1000
start_date = pd.Timestamp("2022-01-01")
end_date = start_date +
pd.DateOffset(days=num_samples - 1)
date_range = pd.date_range(start=start_date,
end=end_date, freq='D')

# Simulate various factors (e.g., CPU usage, number
of active servers, time of day)
cpu_usage = np.random.uniform(low=20, high=80,
size=num_samples)
active_servers = np.random.randint(low=50, high=200,
size=num_samples)
time_of_day = np.sin(np.linspace(0, 2 * np.pi,
num_samples)) * 50 + 50  # Periodic pattern

# Simulate power consumption based on factors
power_consumption = 0.5 * cpu_usage + 0.3 *
active_servers + 0.2 * time_of_day +
np.random.normal(0, 5, size=num_samples)

# Create a DataFrame with synthetic data
data = pd.DataFrame({
    'Date': date_range,
    'CPU Usage': cpu_usage,
    'Active Servers': active_servers,
    'Time of Day': time_of_day,
    'Power Consumption': power_consumption
})

# Split the data into training and testing sets (80%
training, 20% testing)
train_size = int(0.8 * num_samples)
train_data, test_data = data.iloc[:train_size],
data.iloc[train_size:]

# Save the synthetic data to CSV files (optional)
train_data.to_csv('train_data.csv', index=False)
test_data.to_csv('test_data.csv', index=False)
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt

# Extract features and target variable for training
X_train = train_data[['CPU Usage', 'Active Servers',
'Time of Day']]
y_train = train_data['Power Consumption']

# Initialize and train the linear regression model
linear_model = LinearRegression()
linear_model.fit(X_train, y_train)

# Predict on the training set
train_predictions = linear_model.predict(X_train)

# Evaluate the model on the training set
train_rmse = np.sqrt(mean_squared_error(y_train,
train_predictions))
print(f'Training RMSE: {train_rmse}')

# Visualize actual vs predicted values on the
training set
plt.figure(figsize=(12, 6))
plt.plot(train_data['Date'], y_train, label='Actual
Power Consumption', marker='o')
plt.plot(train_data['Date'], train_predictions,
label='Predicted Power Consumption', marker='o')
plt.title('Actual vs Predicted Power Consumption
(Training Set)')
plt.xlabel('Date')
plt.ylabel('Power Consumption')
plt.legend()
plt.show()

# Extract features and target variable for testing
X_test = test_data[['CPU Usage', 'Active Servers',
'Time of Day']]
y_test = test_data['Power Consumption']
# Predict on the test set
test_predictions = linear_model.predict(X_test)
# Evaluate the model on the test set
test_mse = mean_squared_error(y_test,
test_predictions)
test_r_squared = linear_model.score(X_test, y_test)
print(f'Test MSE: {test_mse}')
print(f'Test R-squared: {test_r_squared}')
# Visualize actual vs predicted values on the test
set
plt.figure(figsize=(12, 6))
plt.plot(test_data['Date'], y_test, label='Actual
Power Consumption', marker='o')
plt.plot(test_data['Date'], test_predictions,
label='Predicted Power Consumption', marker='o')
plt.title('Actual vs Predicted Power Consumption
(Test Set)')
plt.xlabel('Date')
plt.ylabel('Power Consumption')
plt.legend()
plt.show()

import numpy as np
```

```
# Function to dynamically adjust prediction
thresholds
def adjust_threshold(predictions, time_of_day,
peak_threshold):
    # Adjust threshold based on peak vs. off-peak
hours
    is_peak_hour = time_of_day > peak_threshold
    adjusted_threshold = np.where(is_peak_hour, 10,
5)
    # Apply dynamic threshold
    adjusted_predictions = np.where(predictions >
adjusted_threshold, 1, 0)
    return adjusted_predictions
peak_threshold = 18
# Apply dynamic threshold to test predictions based
on 'Time of Day'
adjusted_test_predictions =
adjust_threshold(test_predictions, test_data['Time
of Day'],peak_threshold)

import matplotlib.pyplot as plt

# Visualize actual vs predicted values on the test
set
plt.figure(figsize=(12, 6))
plt.plot(test_data['Date'], y_test, label='Actual
Power Consumption', marker='o')
plt.plot(test_data['Date'], test_predictions,
label='Predicted Power Consumption', marker='o',
linestyle='dashed')
plt.title('Actual vs Predicted Power Consumption
Over Time')
plt.xlabel('Date')
plt.ylabel('Power Consumption')
plt.legend()
peak_hours = test_data['Time of Day'] >
peak_threshold
plt.fill_between(test_data['Date'][peak_hours], 0,
max(y_test), color='gray', alpha=0.2, label='Peak
Hours')

plt.show()
```

REFERENCES

[1] Alarifi, A., Dubey, K., Amoon, M., Altameem, T., Abd El-Samie, F. E., Altameem, A., ... & Nasr, A. A. (2020). Energy-efficient hybrid framework for green cloud computing. *IEEE Access*, *8*, 115356-115369.

[2] Bindhu, V., & Joe, M. (2019). Green cloud computing solution for operational cost efficiency and environmental impact reduction. *Journal of ISMAC*, *1*(02), 120-128.

[3] Xu, L., Li, C., Li, L., Liu, Y., Yang, Z., & Liu, Y. (2014, June). A virtual data center deployment model based on green cloud computing. In *2014 IEEE/ACIS 13th International Conference on Computer and Information Science (ICIS)* (pp. 235-240). IEEE.

[4] Wadhwa, B., & Verma, A. (2014, August). Energy and carbon efficient VM placement and migration technique for green cloud datacenters. In *2014 Seventh international conference on contemporary computing (IC3)* (pp. 189-193). IEEE.

[5] Fernández-Cerero, D., Fernández-Montes, A., & Jakóbik, A. (2020). Limiting global warming by improving data-centre software. *IEEE Access*, *8*, 44048-44062.

[6] Doss, R., Gupta, S., Chakravarthi, M. K., Channi, H. K., Koti, A. V., & Singh, P. (2022, April). Understand the Application of Efficient Green Cloud Computing Through Micro Smart Grid in Order to Power Internet Data Center. In *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)* (pp. 336-340). IEEE.

[7] Kumar, R., & Khan, M. A. (2022, December). Analysis of the Effects to Recognize the use of Effective Green Cloud-Based Solutions Using Tiny Smart Grid to Light Web Data Centres. In *2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART)* (pp. 1165-1170). IEEE.

[8] Dey, S., & Pal, S. (2023, May). Controlling Air Pollution in Data Centers using Green Data Centers. In *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing Workshops (CCGridW)* (pp. 200-205). IEEE.

[9] Ye, H., Song, Z., & Sun, Q. (2014, May). Design of green data center deployment model based on cloud computing and TIA942 heat dissipation standard. In *2014 IEEE Workshop on Electronics, Computer and Applications* (pp. 433-437). IEEE.

[10] Wajid, U., Pernici, B., & Francis, G. (2013, October). Energy efficient and CO2 aware cloud computing:

Requirements and case study. In *2013 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 121-126). IEEE.

[11] Hassan, M. B., Saeed, R. A., Khalifa, O., Ali, E. S., Mokhtar, R. A., & Hashim, A. A. (2022, May). Green Machine Learning for Green Cloud Energy Efficiency. In *2022 IEEE 2nd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)* (pp. 288-294). IEEE.