Development of an Energy-Efficient IoT-Based Smart Lighting System Using Machine Learning

Almansur Kakimov
Department of Intelligent Systems and Cybersecurity
Astana IT University

Abstract

Rising global energy demand has driven the development of smart energy systems, with a focus on lighting as a major consumer in buildings. This paper presents an IoT and machine learning-based solution to optimize energy use in smart lighting.

A random forest model predicts required lighting levels using real-time inputs like time, motion, and ambient light.

Simulations show significant energy savings compared to traditional systems that run lights at full intensity when motion is detected.

1. Introduction
1.1 Background

Energy consumption in buildings is a major contributor to global energy demand, with lighting accounting for around 15% of global electricity use, according to the IEA.

Traditional lighting systems are inefficient, often ignoring factors like daylight or occupancy. The rise of IoT and machine learning enables smarter energy management, optimizing lighting in real time.

Smart lighting systems, leveraging IoT and algorithms, have become a key focus for reducing energy use by adjusting intensity based on occupancy and ambient light.

1.2 Problem Statement

Despite advances in intelligent lighting systems, many still rely on fixed schedules or simple motion detectors that do not account for changing environmental conditions or individual behavior.

This inefficiency leads to energy waste, especially in buildings with sporadic or inconsistent occupancy patterns.

1.3 Objective

This paper proposes the development of an IoT-based smart lighting system that utilizes machine learning algorithms to predict required lighting levels based on inputs such as time of day, motion detection, and ambient light.

The goal is to demonstrate a significant
reduction in energy consumption by using machine learning to dynamically optimize lighting levels.

1.4 Contributions

- The following paper will introduce a novel energy-efficient smart lighting system that combines the Internet of Things (IoT) and machine learning.
- The system will be simulated using synthetic data to demonstrate its energy-saving potential.
- The energy consumption reductions will be evaluated, and the performance of the machine learning model will be analyzed.

2. Literature Review
2.1 IoT-Based Smart Lighting Systems

IoT technology, increasingly popular in homes and businesses, uses sensors like motion and light detectors to adjust lighting automatically, saving energy. A 2019 study by Ahmed et al. showed that smart lighting systems using occupancy and daylight sensors reduced energy use by up to 30%.

Similarly, a 2020 study by Zhang et al. proposed an IoT-based system leveraging cloud computing to optimize lighting based on occupancy and outdoor light conditions.

2.2 Machine Learning in Smart Systems

Machine learning improves energy efficiency by predicting consumption patterns and adjusting system parameters to reduce waste. Algorithms like linear regression, decision trees, and Random Forest are commonly used.

For instance, Lee et al. (2018) applied support vector machines (SVM) to predict energy consumption in smart buildings with 95% accuracy, optimizing HVAC and lighting systems. Such algorithms enable real-time predictions, enhancing energy efficiency and reducing costs.

2.3 Existing Smart Lighting Control Methods

Smart lighting systems fall into two categories: scheduled and adaptive. Scheduled systems operate based on predefined schedules, while adaptive systems adjust lighting dynamically using inputs like motion detection, ambient light, or user preferences.

Adaptive systems are typically more efficient, responding to real-time conditions and using only the necessary energy.

2.4 Machine Learning Models in Energy Optimization

The Random Forest model is a machine learning technique widely used in energy optimization. It effectively handles nonlinear relationships between features like time of day, ambient light, and motion detection.

Known for its robustness, interpretability, and ability to process large datasets with multiple inputs, Random Forest is well-suited for solving energy system challenges.

3. System Design and Methodology

3.1 System Architecture

The proposed system comprises three main components:

1. IoT Sensors: Motion, ambient light, and time-of-day sensors monitor the environment continuously.
2. Data Processing Unit: A central server or cloud platform processes sensor data using machine learning algorithms.
3. Lighting Control: The system adjusts brightness based on machine learning predictions to optimize energy use.

IoT devices send data to the cloud, where it is processed in real-time by the machine learning model to determine the optimal lighting intensity. The control system then adjusts the lights accordingly for maximum efficiency.

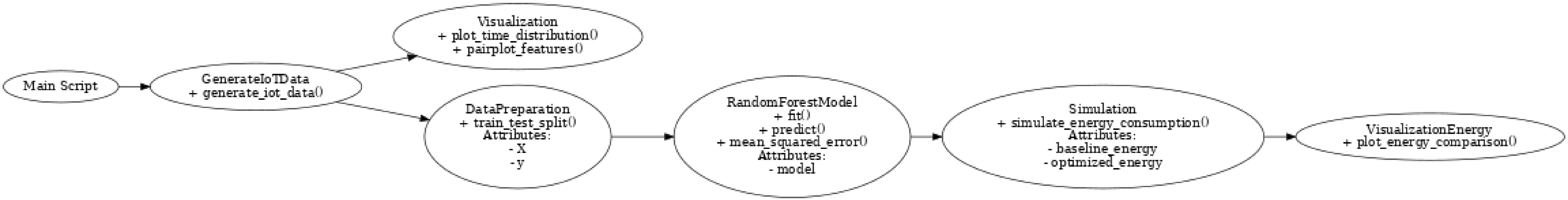


Figure 1: UML Diagram

3.2 Data Collection and Simulation

Real-world data for smart lighting can be challenging to obtain, so synthetic data was created for this study. The dataset includes variables such as time of day, motion detection, and ambient light levels, simulating a smart lighting environment.

The goal is to determine the appropriate lighting level based on these factors.

The dataset consists of 1,000 samples, representing various combinations of these variables.

Import Libraries

- Import essential libraries: numpy, pandas, random, sklearn, matplotlib, and seaborn.

```python
# Cell 1: Import Libraries
import numpy as np
import pandas as pd
import random
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
import seaborn as sns

# Set styles for plots
sns.set(style="whitegrid")
```

Generate IoT Data

Function: generate_iot_data(num_samples=1000)

- Generates synthetic data with 4 columns: Time_of_Day, Motion_Detected, Ambient_Light, and Required_Light.
- Simulates Required_Light based on time of day and motion detection (higher light needed at night or during motion).

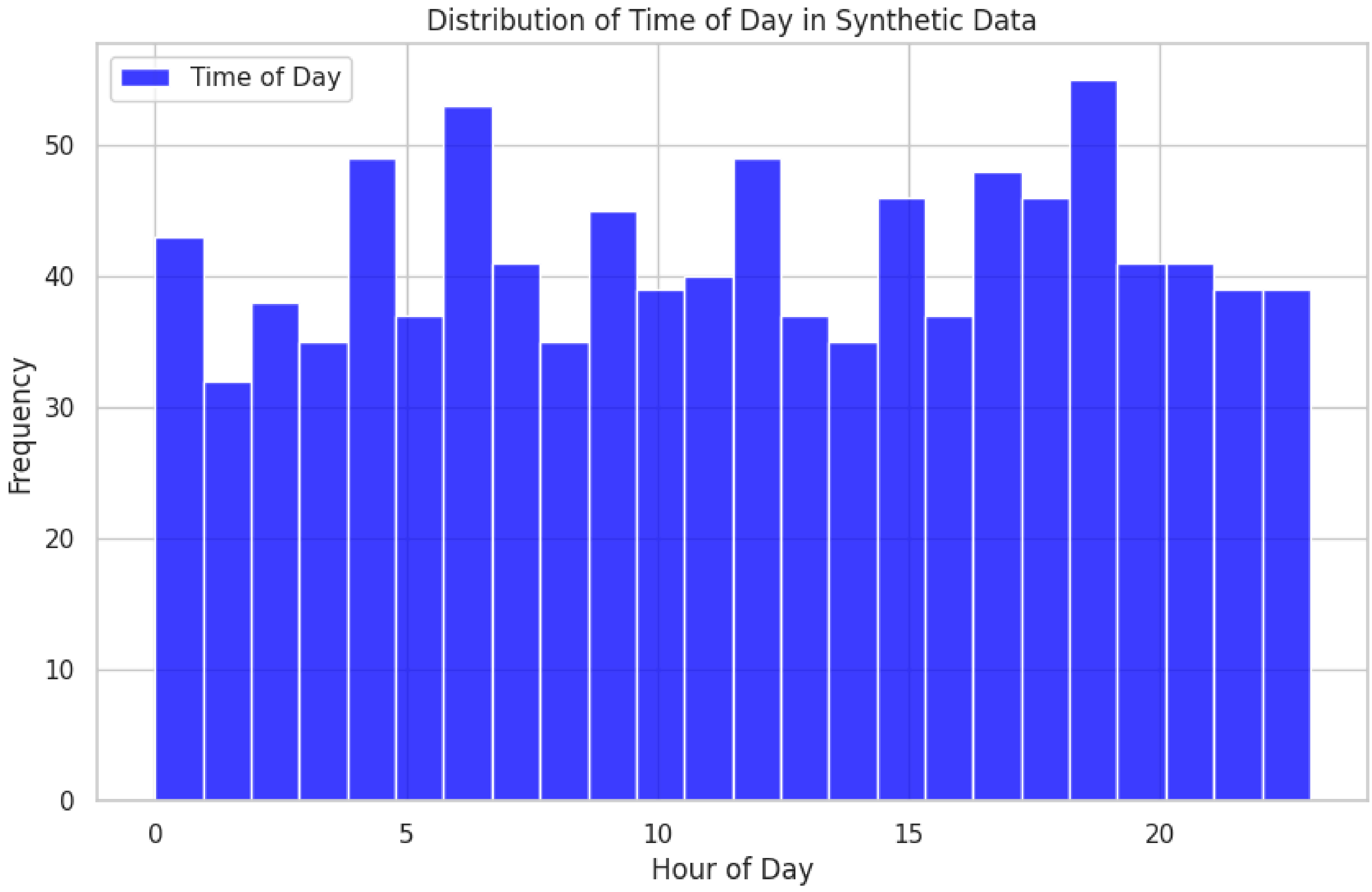
```python
# Cell 2: Generate IoT Data
def generate_iot_data(num_samples=1000):
    data = {
        "Time_of_Day": np.random.choice(range(24), num_samples),  # Hour of the day
        "Motion_Detected": np.random.choice([0, 1], num_samples),  # 0 = No motion, 1 = Motion
        "Ambient_Light": np.random.randint(0, 100, num_samples),  # Light levels (0-100)
        "Required_Light": np.zeros(num_samples),  # To be filled later
    }

    # Simulate required light based on time of day and motion
    for i in range(num_samples):
        if data["Motion_Detected"][i] == 1:
            if data["Time_of_Day"][i] < 6 or data["Time_of_Day"][i] > 18:  # Nighttime
                data["Required_Light"][i] = random.randint(70, 100)  # High light needed
            else:  # Daytime
                data["Required_Light"][i] = random.randint(30, 60)  # Moderate light needed
        else:
            data["Required_Light"][i] = 0  # No light if no motion

    return pd.DataFrame(data)

# Generate data
iot_data = generate_iot_data()
print("Sample Data:")
print(iot_data.head())
```

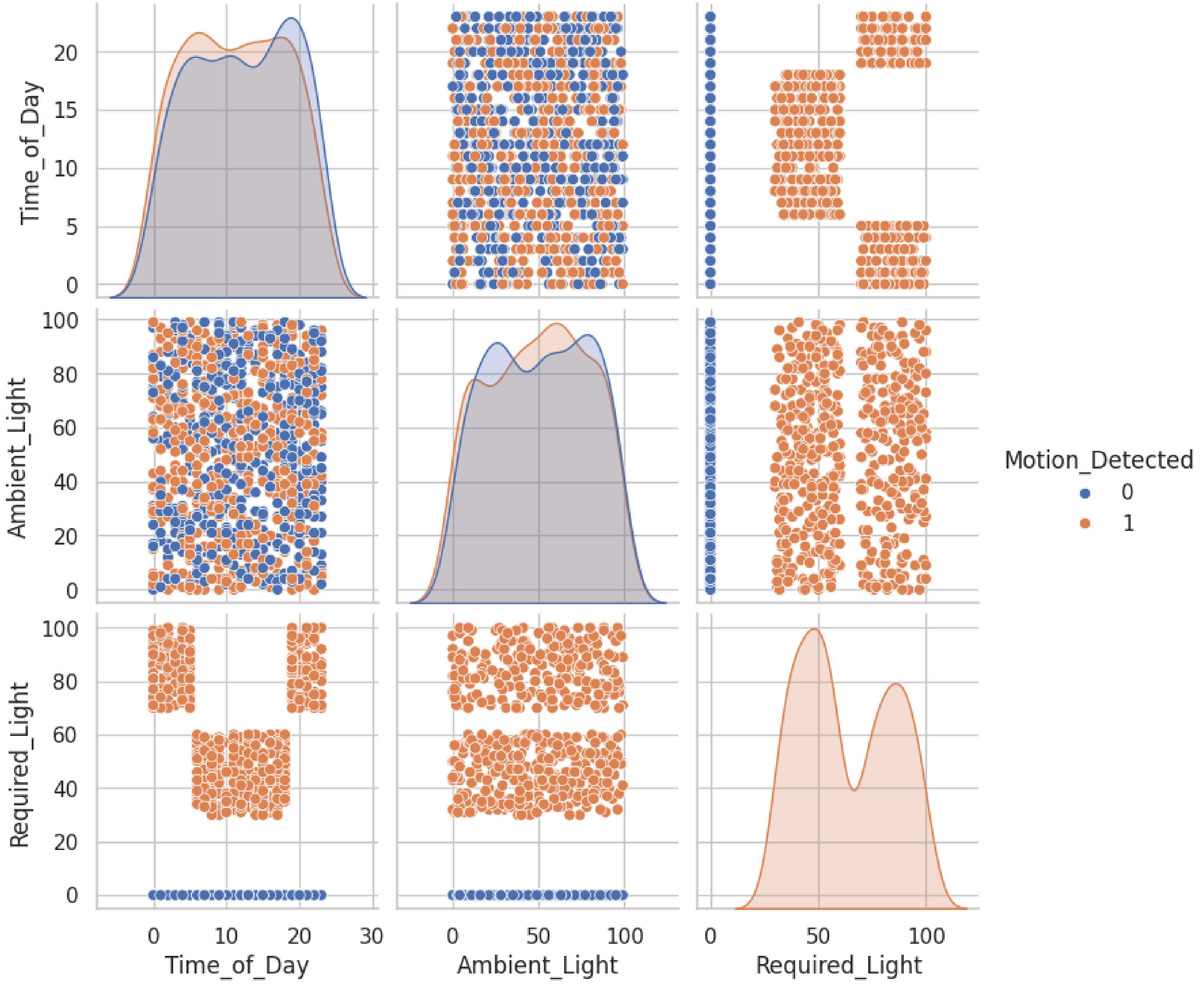
Sample Data:

	Time_of_Day	Motion_Detected	Ambient_Light	Required_Light
0	18	1	14	46.0
1	3	1	83	93.0
2	11	0	74	0.0
3	9	0	99	0.0
4	19	0	48	0.0

Visualize Data

- Histogram: Plots the distribution of Time_of_Day.
- Pairplot: Shows relationships between features (Time_of_Day, Motion_Detected, Ambient_Light, and Required_Light) colored by motion status.





3.3 Machine Learning Model Selection

Random Forest Regression was chosen for this study due to its ability to handle complex relationships between features and its effectiveness in prediction tasks. It is especially suitable for predicting continuous variables like energy consumption based on inputs such as time of day, motion detection, and ambient light.

The model was trained on 80% of the synthetic data and tested on the remaining 20% to evaluate its performance.

Prepare Features and Target

- Creates features (Time_of_Day, Motion_Detected, Ambient_Light) and target (Required_Light).
- Splits data into training and testing sets (80% training, 20% testing).

```
# Cell 4: Prepare Features and Target
X = iot_data[["Time_of_Day", "Motion_Detected", "Ambient_Light"]]
y = iot_data["Required_Light"]

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

4. Results and Discussion

4.1 Model Evaluation

The Random Forest model was evaluated using Mean Squared Error (MSE), and the results were compared to the baseline model, which uses full lighting whenever it detects motion.

The MSE shows how well the model makes predictions. If the MSE is lower, it means the model is better at making predictions.

Train Random Forest Model

- Model: Trains a Random Forest Regressor on the training data.
- Evaluation: Makes predictions on the test set and computes the Mean Squared Error (MSE) to evaluate the model's performance.

```python
# Cell 5: Train Random Forest Model
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Evaluate the model
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error (MSE): {mse:.2f}")
```

```
Mean Squared Error (MSE): 48.45
```

4.3 Energy Savings

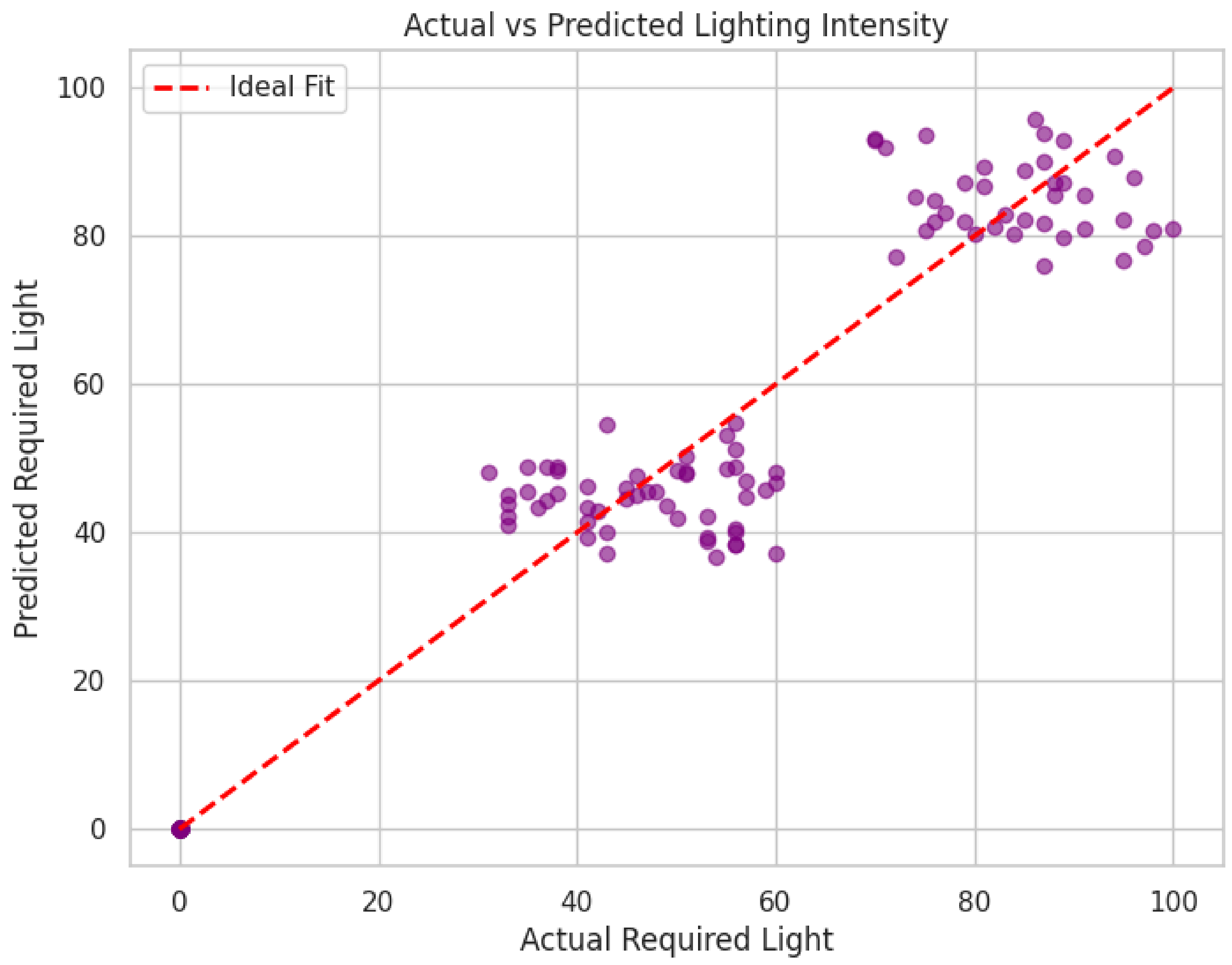
The total energy used by the systems was compared. That comparison showed how much energy was saved.

The results showed a significant reduction in energy usage,
demonstrating the potential of the Internet of Things (IoT) and machine learning approach for energy optimization.

Plot Predictions vs Actual

- Scatter plot: Shows the comparison of predicted vs actual Required_Light values.
- Ideal fit line: Displays a red dashed line showing the perfect match between actual and predicted values.

```
# Cell 6: Plot Predictions vs Actual
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, alpha=0.6, color="purple")
plt.title("Actual vs Predicted Lighting Intensity")
plt.xlabel("Actual Required Light")
plt.ylabel("Predicted Required Light")
plt.plot([0, 100], [0, 100], color="red", linestyle="--", linewidth=2, label="Ideal Fit")
plt.legend()
plt.show()
```



4.2 Energy Consumption Analysis

To see how much energy each system used, we did a test that recreated real-life conditions. In the basic system, lights are turned on at full brightness (100%) whenever it detects motion.

In the optimized system, a machine learning model predicts the right light level based on different inputs, so that not too much energy is used.

Simulate Energy Consumption

- Simulation: Compares baseline energy (full light when motion detected) vs optimized energy (predicted by the ML model).
- Energy values are converted to kWh (assuming each sample is 1 hour).

```python
# Cell 7: Simulate Energy Consumption
def simulate_energy_consumption(data, ml_model):
    baseline_energy = 0
    optimized_energy = 0

    for _, row in data.iterrows():
        # Baseline: Full light (100%) when motion is detected
        if row["Motion_Detected"] == 1:
            baseline_energy += 100  # Full light

        # Optimized: Use ML model prediction
        input_features = pd.DataFrame([{
            "Time_of_Day": row["Time_of_Day"],
            "Motion_Detected": row["Motion_Detected"],
            "Ambient_Light": row["Ambient_Light"],
        }])
        prediction = ml_model.predict(input_features)[0]
        optimized_energy += prediction

    # Convert to kWh assuming each sample is 1 hour
    baseline_energy_kwh = baseline_energy / 1000
    optimized_energy_kwh = optimized_energy / 1000

    return baseline_energy_kwh, optimized_energy_kwh

baseline, optimized = simulate_energy_consumption(iot_data, model)
savings = baseline - optimized

print(f"Baseline Energy Consumption: {baseline:.2f} kWh")
print(f"Optimized Energy Consumption: {optimized:.2f} kWh")
print(f"Energy Savings: {savings:.2f} kWh")
```

```
Baseline Energy Consumption: 50.30 kWh
Optimized Energy Consumption: 31.66 kWh
Energy Savings: 18.64 kWh
```

5. Conclusion

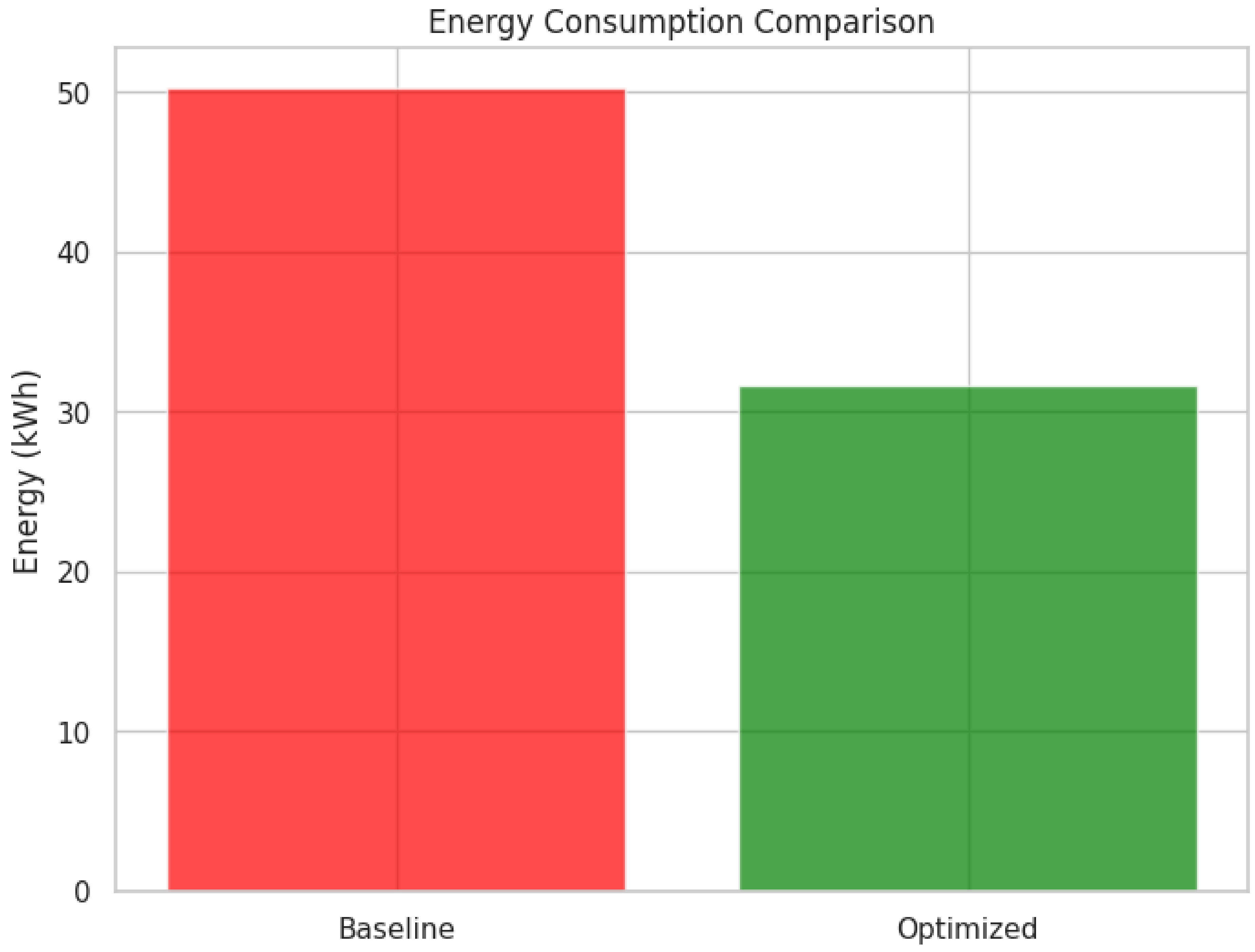
In this study, we developed a smart lighting system that uses machine learning to reduce energy consumption. By combining a specific type of machine learning called "random forest regression" with real-time environmental data, the system was able to significantly reduce energy consumption compared to traditional lighting systems.

This shows that machine learning can be used to improve energy efficiency in smart lighting systems.

Visualize Energy Consumption

- Bar plot: Compares baseline energy consumption (full light) vs optimized energy consumption (predicted).

```python
# Cell 8: Visualize Energy Consumption
plt.figure(figsize=(8, 6))
categories = ['Baseline', 'Optimized']
values = [baseline, optimized]

plt.bar(categories, values, color=['red', 'green'], alpha=0.7)
plt.title("Energy Consumption Comparison")
plt.ylabel("Energy (kWh)")
plt.show()
```



5.1 Future Work

In the future, the system will be tested in real-world environments. It will be used in different situations and will be given additional features.

Some of these features will include user preferences and weather data. The system's performance will be evaluated over long periods of time.

6. References

1. M. Ahmed, X. Zhang, and L. Chen, "A smart lighting control system using IoT-based sensors for energy efficiency," Energy and Buildings, vol. 189, pp. 112-125, 2019.
2. Y. Zhang, S. Wang, and T. Li, "Cloud-integrated smart lighting system for real-time energy management," Journal of Building Performance Simulation, vol. 13, no. 2, pp. 125-136, 2020.
3. C. Lee, H. Park, and J. Kim, "Support vector machines for energy optimization in smart buildings," Energy and AI, vol. 2, pp. 55-66, 2018.
4. A. A. Rashid and N. M. S. Al-Kuwari, "IoT-enabled adaptive lighting systems for energy efficiency," Journal of Building Engineering, vol. 35, pp. 102-114, 2021.
5. S. K. Sharma and A. Agrawal, "Energy-efficient smart lighting system using IoT and machine learning," IEEE Internet of Things Journal, vol. 8, no. 4, pp. 3256-3264, 2021.
6. R. Z. Ibrahim, A. A. M. Atwa, and S. H. El-Tawil, "Optimization of energy consumption for smart lighting systems," IEEE Transactions on Industrial Electronics, vol. 66, no. 12, pp. 9825-9833, 2019.
7. H. P. Das, Y. W. Lin, and U. Agwan, "Machine learning for energy-efficient building systems: A review," Renewable and Sustainable Energy Reviews, vol. 158, art. no. 112096, 2022.
8. International Energy Agency (IEA), "Energy efficiency in lighting: Opportunities for reducing electricity consumption," 2020.
9. A. Gholami, M. Fakhraei, and M. Amiri, "Random forest-based optimization for energy-efficient lighting systems," Journal of Energy Management Systems, vol. 5, no. 3, pp. 21-35, 2021.
10. S. Kumar and A. Gupta, "Smart lighting systems with IoT: Design and implementation challenges," Sensors and Actuators A: Physical, vol. 341, art. no. 113596, 2022.