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

Almansur Kakimov
Department of Intelligent Systems and Cybersecurity
Astana IT University
221181@astanait.edu.kz

Abstract

The increasing global demand for energy has prompted the development of smart energy systems, especially in lighting, which is one of the largest energy consumers in buildings. To optimize energy consumption in smart lighting systems, this paper proposes a solution based on the Internet of Things (IoT) and machine learning (ML). Using a random forest model, we predict the required lighting level based on real-time inputs such as time of day, motion detection, and ambient light. Our simulation demonstrates a significant reduction in energy consumption from an optimized system compared to the traditional baseline system, which operates lights at full intensity whenever motion is detected.

Keywords: *Smart Lighting, Energy Efficiency, Machine Learning, IoT, Random Forest, Energy Consumption, Smart Systems*

1. Introduction

1.1 Background

Energy consumption in buildings accounts for a significant portion of global energy demand, and lighting is one of the largest consumers of electricity. According to the International Energy Agency (IEA), lighting alone accounts for approximately 15% of global electricity consumption. Traditional lighting systems are often inefficient, using the same amount of energy regardless of environmental factors such as daylight availability or human activity.

With the advent of the Internet of Things (IoT), the potential for smarter energy management has grown. IoT-based systems, when combined with machine learning algorithms, offer the opportunity to optimize energy use by adjusting environmental factors in real time.

In recent years, there has been a significant focus on developing energy-efficient solutions for building management systems. Smart lighting systems have emerged as a key area where IoT devices and machine learning can significantly reduce energy consumption by dynamically adjusting lighting intensity based on occupancy and ambient light levels.

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

In recent years, Internet of Things (IoT) technology has become popular in homes and businesses. This technology uses sensors like motion and light detectors to adjust lighting automatically. Studies have shown that these systems can save energy by adjusting the lighting based on things like how many people are in a room, the amount of sunlight, and the time of day.

For example, a 2019 study by Ahmed et al. created a smart lighting system that uses occupancy and daylight sensors to adjust lighting in real time, reducing energy use by up to 30%. Another study in 2020 by Zhang et al. proposed a smart lighting system that uses the IoT and cloud computing to control lighting based on occupancy and the outside lighting conditions.

2.2 Machine Learning in Smart Systems

Machine learning has been used a lot to improve energy use in different systems, like lighting. Machine learning models can predict energy consumption patterns and adjust system parameters to minimize waste. There are different algorithms that can be used for this, like linear regression, decision trees, and Random Forest.

One important example is the work of Lee et al. (2018), who used something called support vector machines (SVM) to predict energy consumption in smart buildings. Their model was

95% accurate in predicting energy needs, which helps optimize heating, ventilation, and air conditioning (HVAC) and lighting systems. Machine learning algorithms can make real-time predictions that help people use energy more efficiently and reduce costs.

2.3 Existing Smart Lighting Control Methods

There are two main types of smart lighting systems: scheduled lighting control and adaptive lighting control. Scheduled systems turn lights on or off based on a predefined schedule. Adaptive systems adjust lighting levels based on things like motion detection, ambient light, or user preferences. Adaptive systems are usually more efficient because they respond to changing conditions, like when someone is moving around. This way, they only use as much energy as they need to.

2.4 Machine Learning Models in Energy Optimization

The Random Forest model is a type of machine learning that can solve problems related to energy systems. It can handle nonlinear relationships between features (such as time of day, ambient light, and motion detection) and energy consumption. It's been used a lot in energy optimization because it's strong, easy to understand, and can handle large datasets with many input variables.

3. System Design and Methodology

3.1 System Architecture

The proposed system consists of three main components:

1. IoT Sensors: Motion detection, ambient light, and time-of-day sensors continuously monitor the environment.
2. Data Processing Unit: This is a central server or cloud-based platform that processes data from IoT devices and uses machine learning algorithms.
3. Lighting Control: The lighting system adjusts the brightness based on predictions from the machine learning model, ensuring energy efficiency.

IoT devices communicate with a cloud server where data is processed right away. The machine learning model is used in the cloud to predict the right lighting intensity. This is based on the data that is collected. The control system adjusts the lights based on this prediction.

3.2 Data Collection and Simulation

It's hard to get real-world data for smart lighting, so we made synthetic data for this study. The data includes variables such as time of day, motion detection, and ambient light levels, simulating a smart lighting environment. The main thing we're looking at is the right amount of light, which depends on the time of day and whether something is moving or not.

The generated data simulates 1,000 samples, representing various combinations of time of day, motion detection, and ambient light levels.

3.3 Machine Learning Model Selection

This study selected Random Forest Regression for two reasons. First, it can handle complex relationships between features. Second, it is effective for prediction tasks. It is particularly useful for predicting continuous variables, like energy consumption, based on multiple input features such as time of day, motion detection, and ambient light.

The model was trained on 80% of the data and tested on 20% of the data. This process was used to evaluate the model's performance.

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.

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.

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.

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.

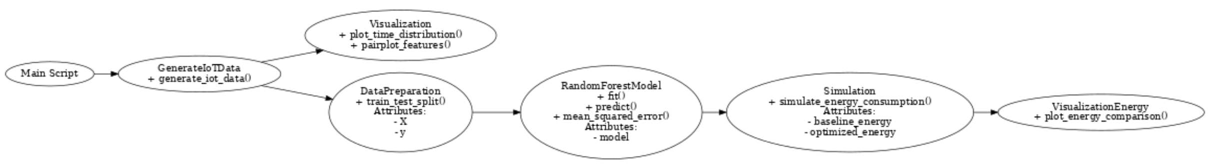
5.1 Future Work

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.

UML Diagram:

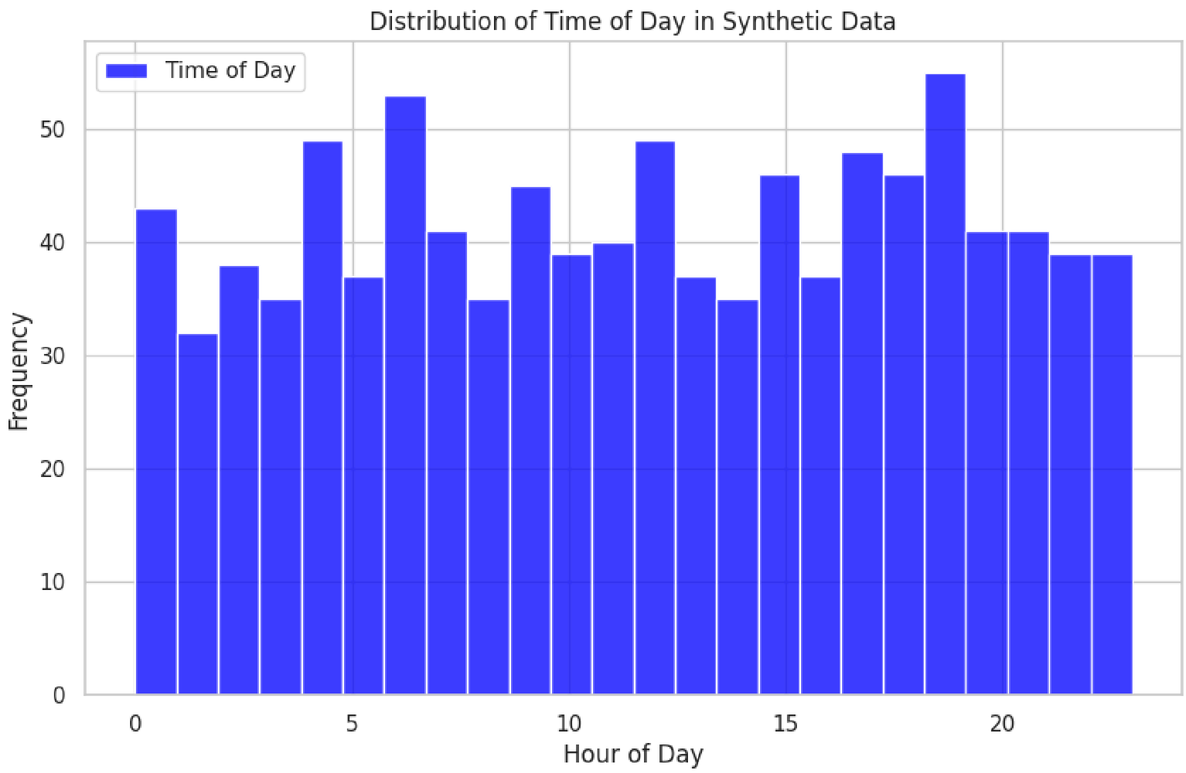


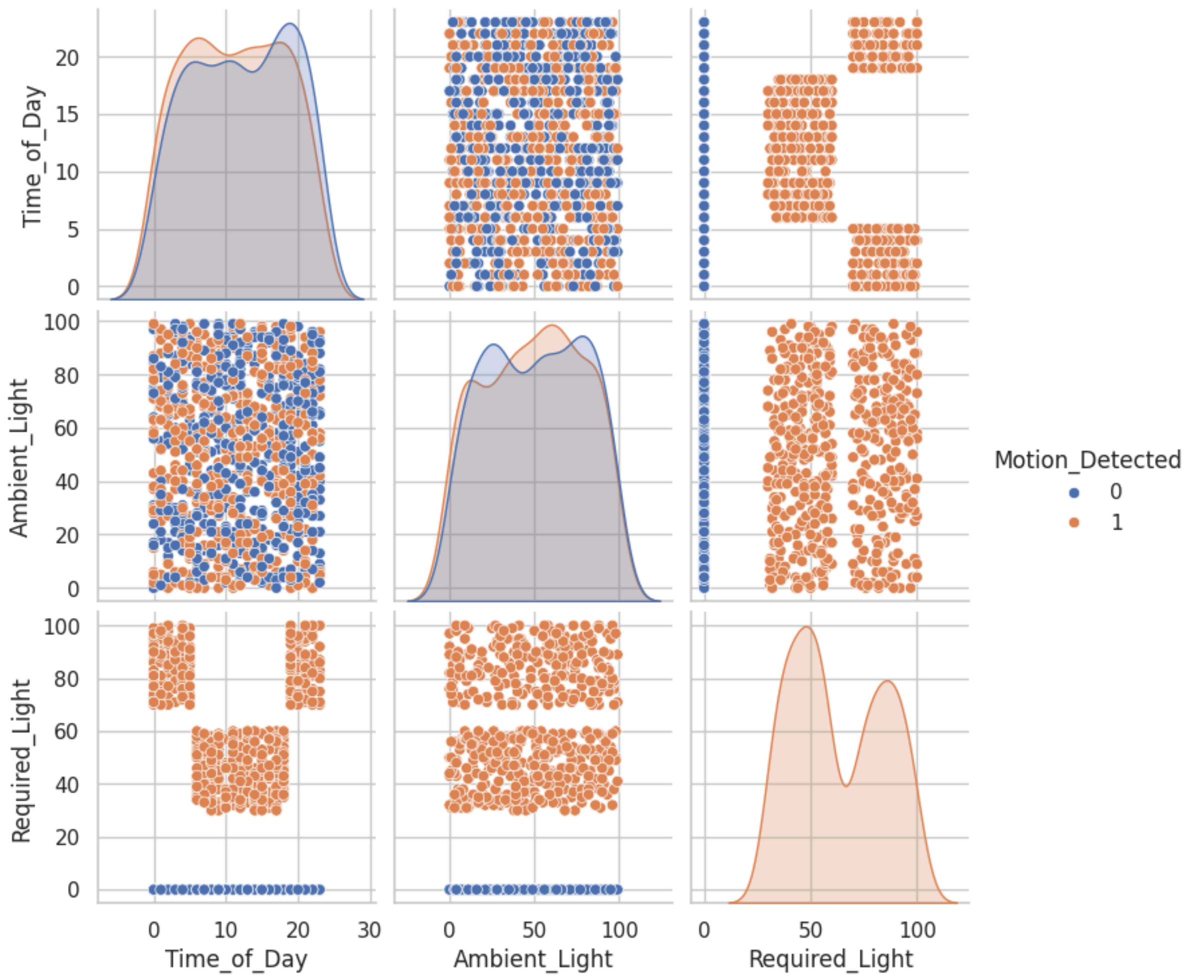

```
# Cell 1: Import Libraries
import numpy as np
import pandas as pd
import random
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
```

```python
import seaborn as sns

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

```python
# Cell 3: Visualize Data
plt.figure(figsize=(10, 6))
sns.histplot(iot_data["Time_of_Day"], kde=False, bins=24, color="blue",
label="Time of Day")
plt.title("Distribution of Time of Day in Synthetic Data")
plt.xlabel("Hour of Day")
plt.ylabel("Frequency")
```

```
plt.legend()
plt.show()

sns.pairplot(iot_data, hue="Motion_Detected")
plt.show()
```



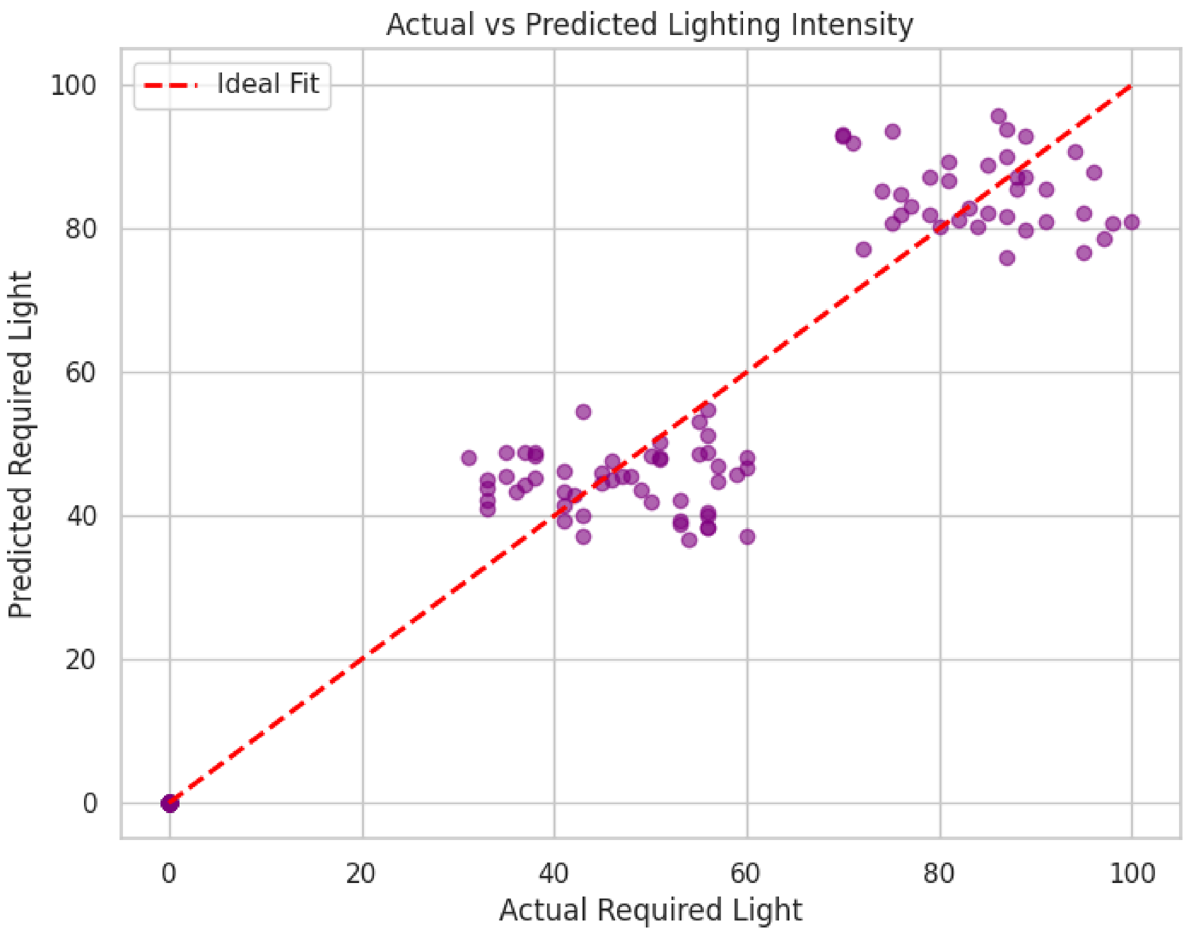


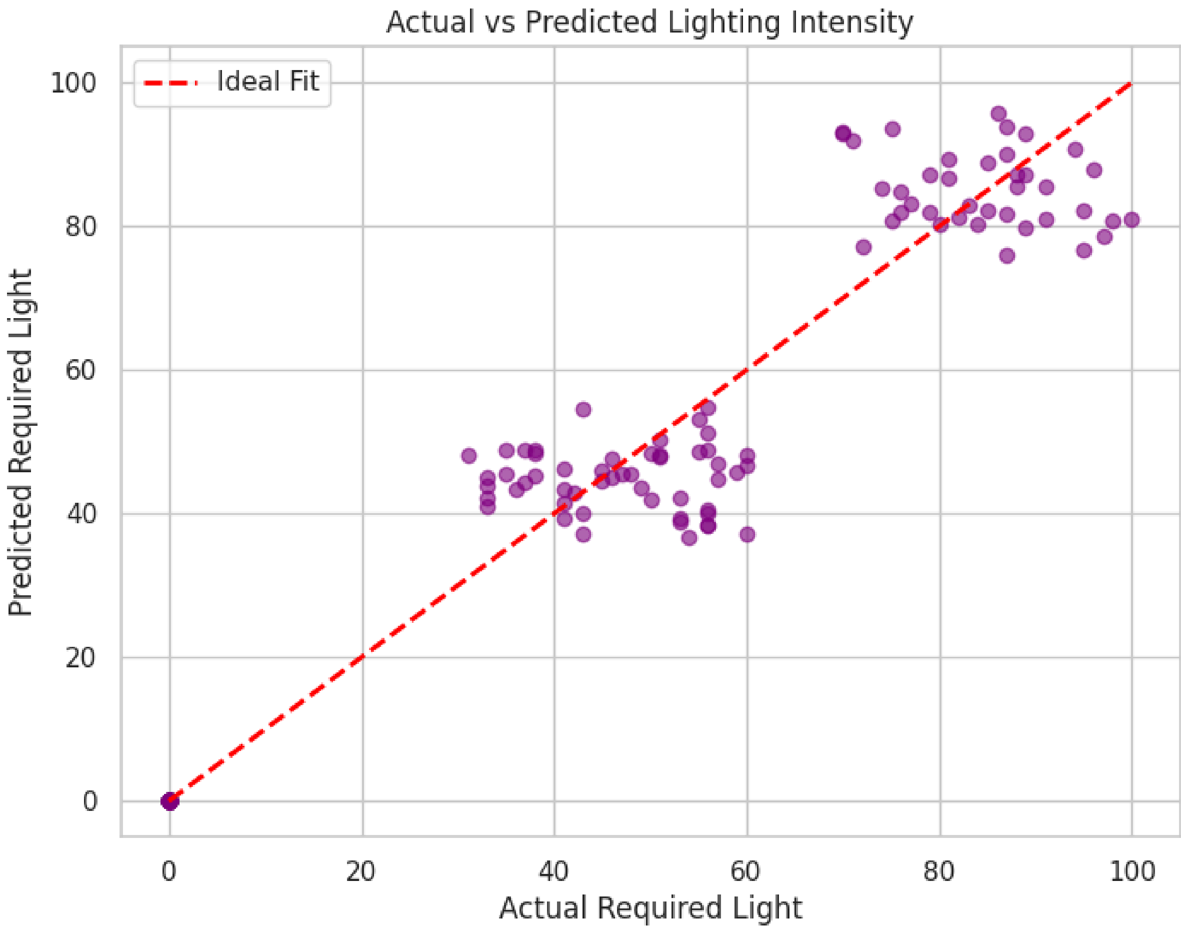

```
# Cell 4: Prepare Features and Target
X = iot_data[["Time_of_Day", "Motion_Detected", "Ambient_Light"]]
y = iot_data["Required_Light"]

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

```
# 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}")
```

```
# Cell 6: Plot Predictions vs Actual
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, alpha=0.6, color="purple")
```

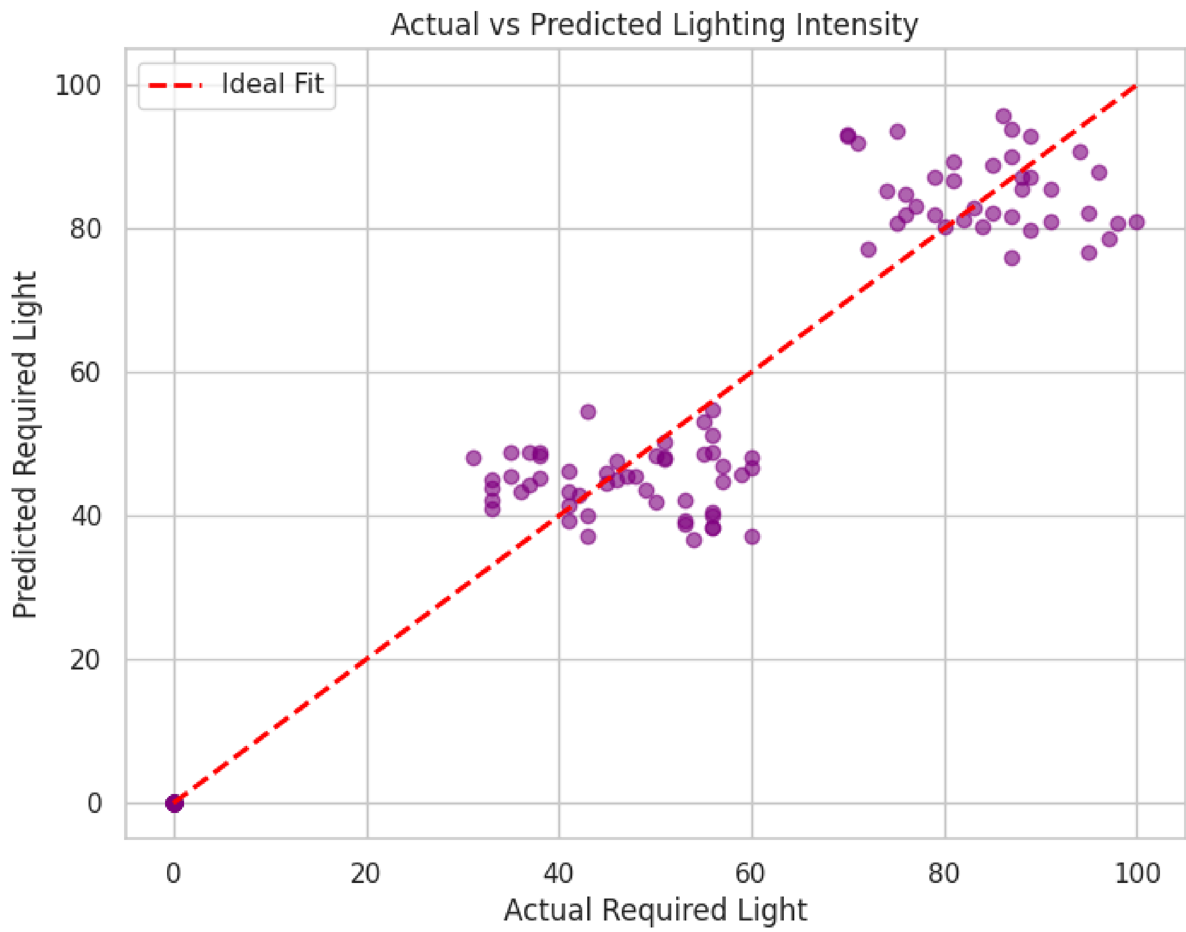
```
plt.title("Actual vs Predicted Lighting Intensity")
plt.xlabel("Actual Required Light")
plt.ylabel("Predicted Required Light")
plt.plot([0, 100], [0, 100], color="red", linestyle="--", linewidth=2,
label="Ideal Fit")
plt.legend()
plt.show()
```



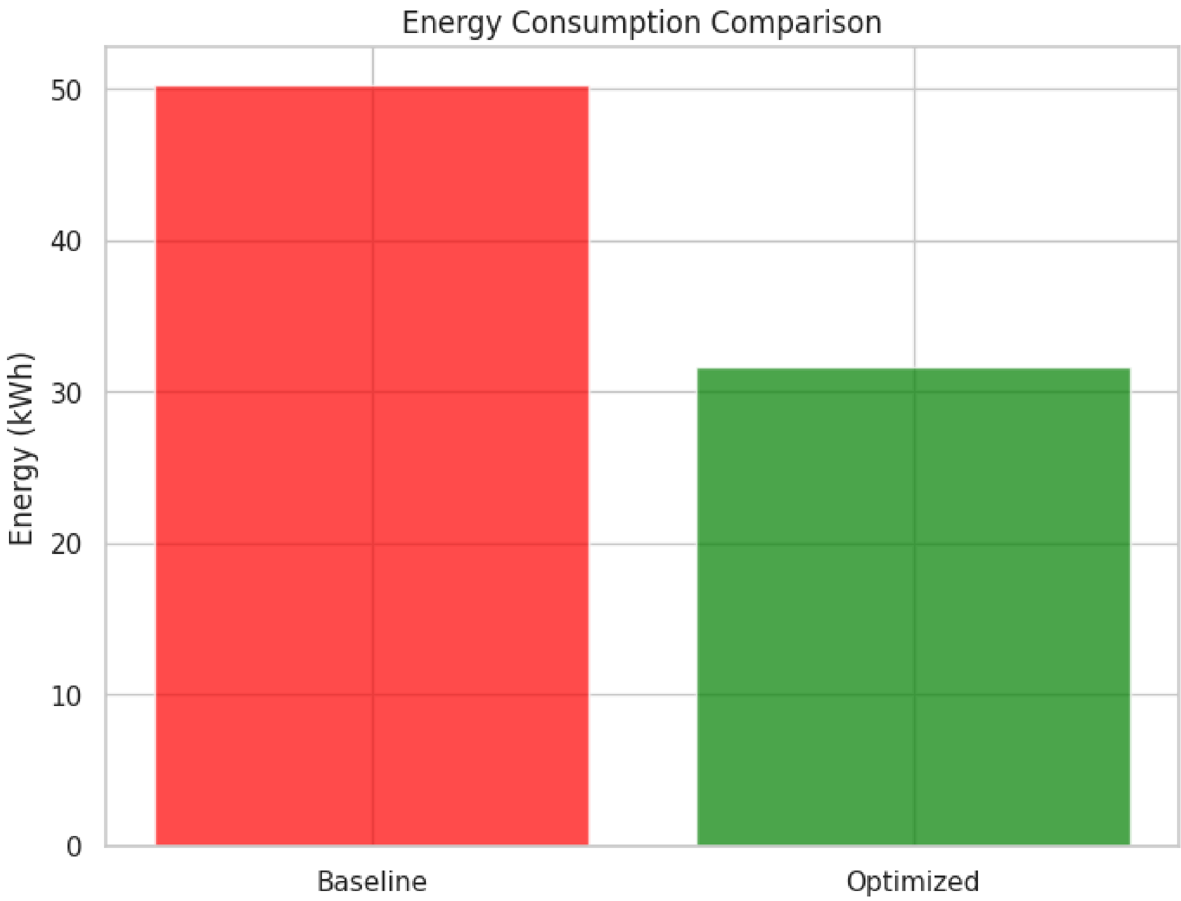



```
# Cell 7: Simulate Energy Consumption
def simulate_energy_consumption(data, ml_model):
   baseline_energy = 0
   optimized_energy = 0

   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")
```




```
# 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()
```


Energy Consumption Comparison
Energy (kWh)
50
40
30
20
10
0
Baseline
Optimized