A Python file that incorporates AI and IoT-based energy efficiency optimization. The script will include key components such as:

1. **Data Collection** – Integrating IoT sensors to gather real-time energy data.
2. **AI-Based Prediction** – Using machine learning models to forecast energy consumption.
3. **Optimization Strategies** – Implementing control mechanisms to reduce energy waste.
4. **Security Measures** – Ensuring secure data transmission and device authentication.

Here's a basic framework for your Python implementation:

import numpy as np

import pandas as pd

from sklearn.ensemble import RandomForestRegressor

import requests

# Simulated IoT data collection

def collect\_energy\_data():

# Replace with actual IoT sensor integration

return {"temperature": np.random.uniform(20, 30), "usage": np.random.uniform(100, 500)}

# AI Model for Energy Prediction

class EnergyPredictor:

def \_\_init\_\_(self):

self.model = RandomForestRegressor(n\_estimators=50)

def train(self, historical\_data):

X = historical\_data[['temperature']]

y = historical\_data['usage']

self.model.fit(X, y)

def predict(self, current\_conditions):

return self.model.predict([current\_conditions])[0]

# Optimization function

def optimize\_energy(predicted\_usage):

threshold = 300 # Example threshold for optimization

if predicted\_usage > threshold:

return "Reduce energy consumption."

else:

return "Maintain current usage."

# Security function for secure communication

def secure\_transmission(data):

# Placeholder for encryption and secure transmission logic

return f"Encrypted Data: {data}"

# Main Execution

historical\_data = pd.DataFrame({'temperature': [22, 24, 26, 28], 'usage': [200, 250, 300, 400]})

predictor = EnergyPredictor()

predictor.train(historical\_data)

current\_conditions = collect\_energy\_data()

predicted\_usage = predictor.predict([current\_conditions["temperature"]])

optimization\_action = optimize\_energy(predicted\_usage)

secure\_data = secure\_transmission(current\_conditions)

print(f"Predicted Usage: {predicted\_usage}")

print(f"Optimization Action: {optimization\_action}")

print(f"Secure Transmission: {secure\_data}")