****

**Boston University**

**Electrical & Computer Engineering**

**EC463 Capstone Senior Design Project**

**Final Testing Plan**

**Machine Learning Powered Electrical Scheduling**



Submitted to

Alan Pisano

8 St. Mary’s Street

Boston, MA

(617) 353 - 6264

apisano@bu.edu

by

|  |
| --- |

Team 28

EcoStrip Solutions

Team Members

Avishai Lean alean@bu.edu

Benjamin Axline baxline@bu.edu

Mateusz Górczak mgorczak@bu.edu

Konstantin Agrachev kostya@bu.edu

Submitted: 04/01/25

#### **Required Materials**

Hardware:

* Kasa Smart Outlet
* Server - (Lab PC)

Software:

* Python 3 Scripts
  + API querying and data formatting scripts
  + Individual Generation Model Scripts
  + Prototype Script to Unify Models
  + Automation Combinations of the models
  + Tunnel for Powerstrip connection
* Javascript Website Design
  + React
  + Node.JS
  + npm
  + FastAPI

#### **Set-Up**

Kasa Smart Plug: We will have the smart plug setup before testing occurs. This process has been greatly simplified and involves for the smart plug to be connected to a router, as well as a script running on a computer connected to that same router.

Data Collection: To begin the data collection process we obtained usernames, passwords, and API keys to update the training CSV file that contains both weather data and power data for ~5-minute intervals. The weather API is <https://www.visualcrossing.com>, while the power data is from <https://www.iso-ne.com>. Similarly, we use these authentications to pull testing (future) data from the API’s.

ML: We will run our projections on this testing data. Once we have updated the data, we will retrain models with the latest information. We then pass these completed models to the prototype function. This function takes in the test data set generated previously and the completed models and calculates the next couple of days of predictions based on these models. The script then writes these predictions to a CSV file energy predictions.

Automation: The process of collecting data, and training machine learning algorithms to match said data has been completely automated. Each day at around 3pm - (shortly before our senior design period) the run\_pull\_push.sh bash file runs. This file pulls all changes from github, performs the data collection, training, and then pushes the resulting CSV file to github. This process is run on a timed basis and there is a scheduled waiting period of one day after pushing the changes concludes.

Website: We read these new energy predictions and generate visualizations based on these outputs. When the CSV file energy prediction is pushed to GitHub, the website displays the predicted generation for hydro, solar, nuclear, and wind in the first line graph. The second line graph displays the predicted generation for refuse and wood. The technologies were divided into two line graphs because the need for refuse and wood is much smaller, therefore they would not be visible on one line graph. The user can also log into the website to control the powerstrip, which can be turned on and off as well as scheduled. This scheduling feature suggests the best times to use an electrical load as a result of the forecasting.

#### **Pre-Testing Setup Procedure**

Smart Plug Side:

1. Connect the Kasa Smart Strip to the Kasa App and choose the MOTODEE2 network.
2. Make sure you have a computer at home connected to the same network.
3. Install ngrok on that same computer: <https://ngrok.com/downloads/>
4. Run ngrok with this command:

ngrok http http://localhost:8080 --url="fast-kid-sterling.ngrok-free.app"

1. Start the powerstrip server with this command:

uvicorn app:app --port 8080 --reload

Machine Learning Side:

1. A CSV File with Training Data from 2021 to the Present has already been created. This file is Auto\_Combine.csv

Website Side (Developer) :

1. Install all necessary dependencies: Node.js, npm, Python3, python3-venv, python-kasa, FastAPI
2. Change the directory to the paper-dashboard
   1. npm install
   2. npm start

Website Side (User):

1. Visit [www.ecostripsolutions.com](http://www.ecostripsolutions.com)
2. Login to the website or create an account.
3. Use website.

#### **Testing Procedure**

Start with ML:

1. Observe the process of how the backend complete function works and sleeps

Website:

1. Login to the website using credentials.
2. Get the latest projections.
3. Display the latest data projections.
4. Turn the outlet on and off.
5. Demonstrate scheduling functionality.

#### 

#### **Measurable Criteria**

The criteria for successful running and output are as follows:

* Successful in gathering data.
* Calculating error for individual ML models to verify results with less than 10% error.
* Ensure that logging into the website successfully occurs.
* Verify data is successfully displayed.
* Verify the website can turn the smart outlet on and off.

#### 

#### **Score Sheet**

| **ML Model** | **% Error** |
| --- | --- |
| Nuclear |  |
| Refuse |  |
| Solar |  |
| Hydroelectric |  |
| Wood |  |
| Wind |  |

| **ML Model** | **Works Correctly?** |
| --- | --- |
| Automation of Data Pulling |  |
| Automation and Correct Format of Output CSV |  |

| **Website** | **Works Correctly?** |
| --- | --- |
| Log In System |  |
| Power Forecasting Display |  |
| Scheduling Works |  |