Capstone Project

Smart Energy Analytic Disaggregation Systems - Platform: An end to end system for energy monitoring of home appliances

Ramin Ebrahimi, Ian Lofgren, Colin Peter, Nikita Sokolnikov

Abstract

Outside of their monthly utility bill, most people do not have a strong grasp on the specifics of their electricity consumption. With time-of-use metering and confusing pricing tiers, managing that cost is not a simple task. SEADS Platform(SEADS-P) is a consumer-friendly **p**latform designed to de-obfuscate household electricity usage so that users can see what appliances are consuming the most energy, and empower consumers to make educated decisions about electricity usage. We use a **N**on-**I**ntrusive **L**oad **M**onitoring (NILM) system to detect the user's power consumption. The SEADS device monitors the aggregate electrical signal of the entire household. SEADS-Platform collects this aggregate data and uses machine learning to disaggregate it into separate signals per appliance. The user can view this information through our web dashboard. Our platform also calculates the cost of the electricity use of each appliance.

Technologies

The website was developed using Django, a Python-driven web framework. Django makes it easy for programmers to build database-driven web applications. This made our front-end and back-end easy to implement, as Django handles both conveniently.

Most of our backend codebase was written in Python. It allowed us to create the system quickly, and easily make changes requested by the client.

Heroku handles all of the build steps necessary for deployment, leaving the developers only responsible for development. Heroku automatically deploys the latest version to the cloud ensuring an always up-to-date website.

PostgreSQL, a relational database engine, was used as the back-end for the Django web application as well as the data store for the time-series data.

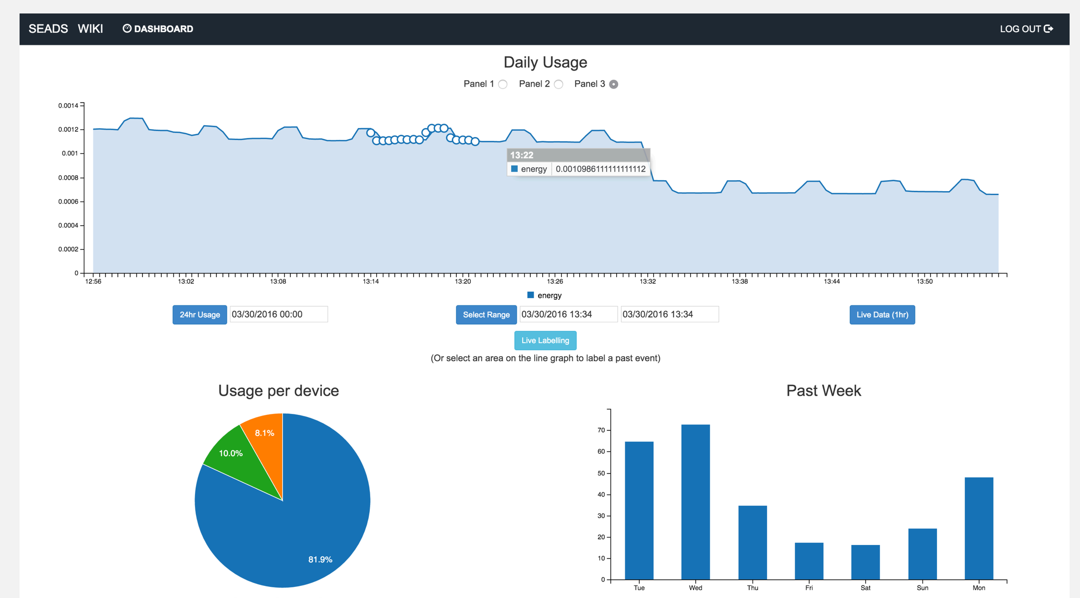
D3 is a JavaScript library for producing dynamic, interactive data visualizations in web browsers. D3 was used for all of the website's data visualization.

Travis CI is an automated building and testing system. Every commit that we pushed to the server got built and tested with Travis CI, and we immediately knew when test coverage was decreased, or when tests/the build failed.

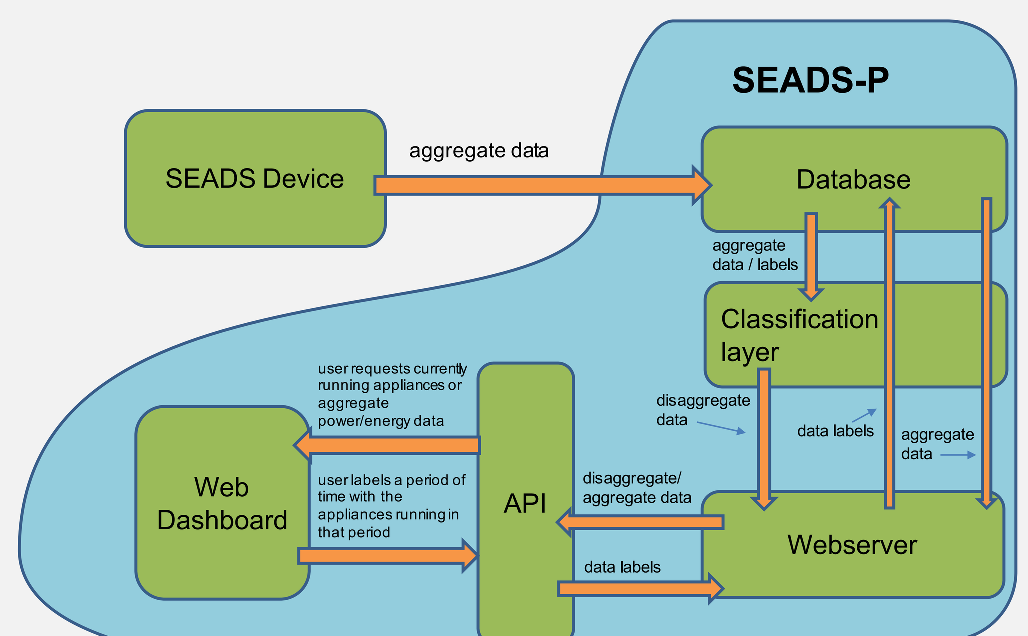
Scikit-learn is a Python machine learning library with support for many classifiers. We use its RandomForestModel for classifying electrical data.

Overview

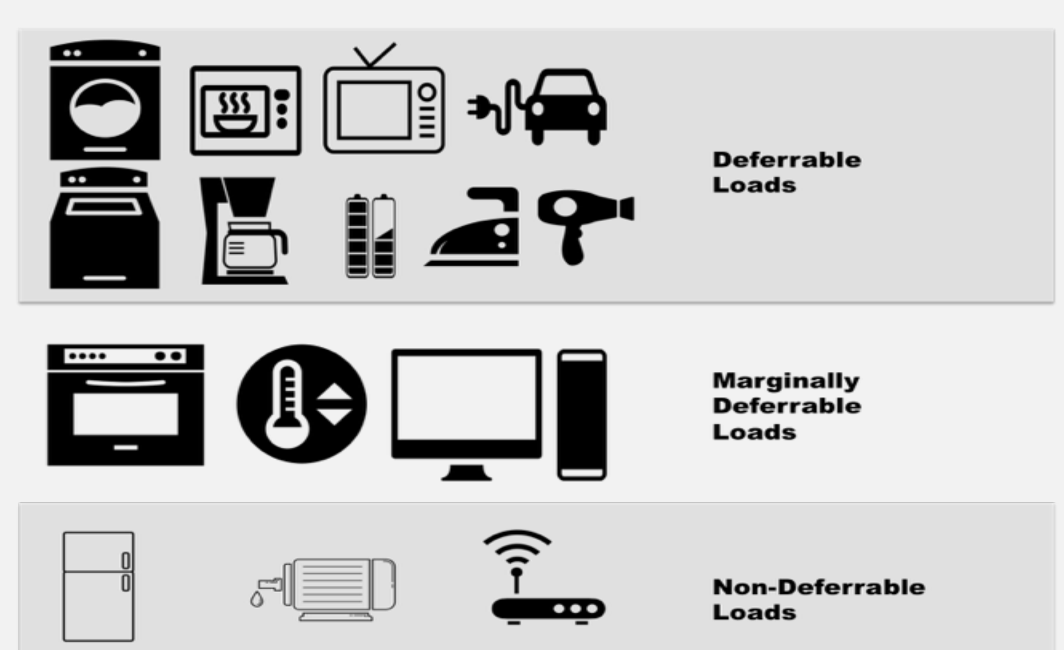
SEADS-P



Dashboard

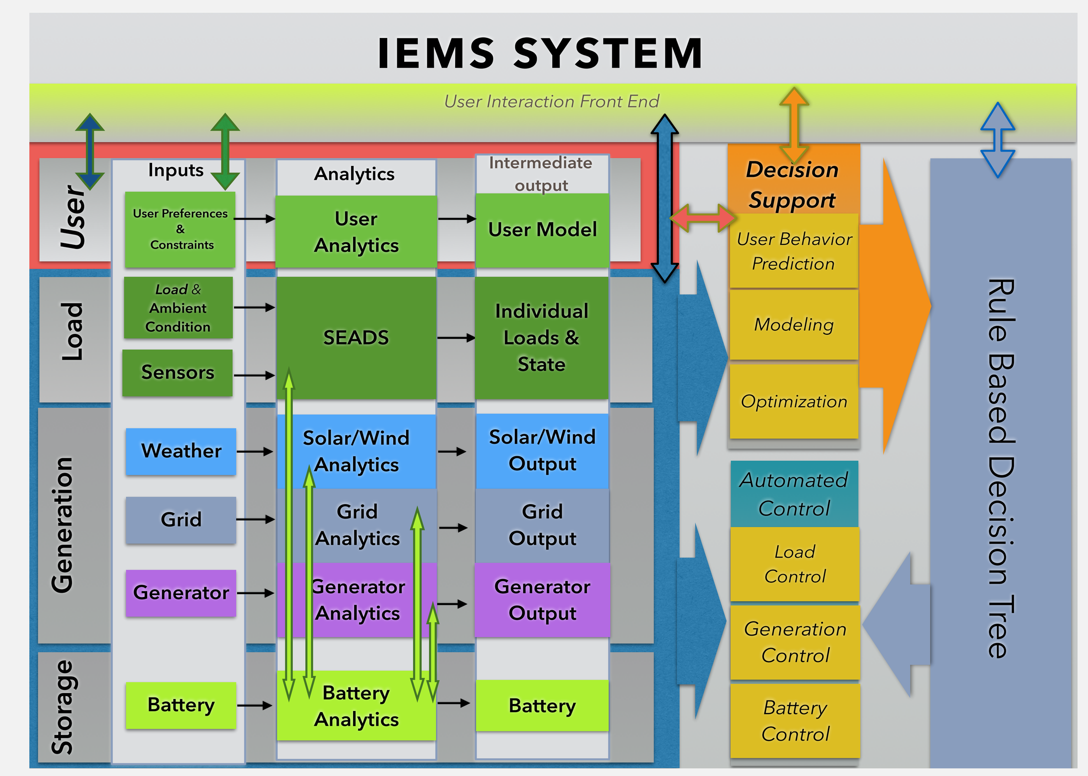


Architecture

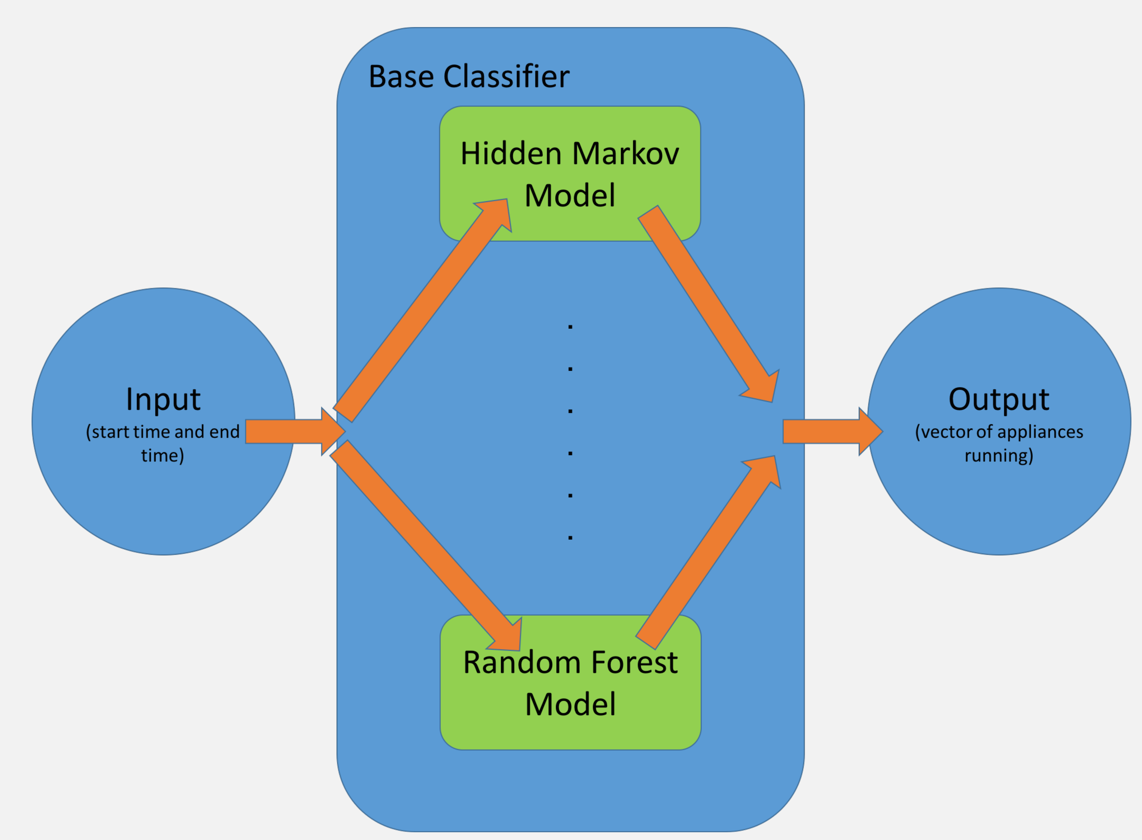


Appliance Categories

IEMS: Intelligent Energy Management System



Disaggregation



The SEADS platform allows integration with established disaggregation algorithms and provides a common interface for testing them against one another.

Implementation

We were tasked with integrating predictive capabilities into the existing SEADS system, which only collected data. To do this, we implemented a classification component that allows using any amount of different disaggregation algorithms to judge and compare their effectiveness. To do this we abstracted the gathering of training data to a base classifier class. This approach provides a common interface to the database that makes creating new classifiers much simpler. The classifiers are all implemented in python, and one of them uses scikit-learn's RandomForestModel to make predictions. An arbitrary time frame, supplied by the user requesting disaggregated data, is used to query the raw data and select data in that time frame. These data are used as input to the RandomForestModel. Multiple appliances running at once are converted into a single bitstring by the base classifier to classify, allowing our data to be used with almost any machine learning technique.

The predictions that the SEADS-P provides are available via a public API. By providing a start and end time, SEADS-P will respond with a vector of currently running appliances and their energy usage in that time. These requests are routed through a request layer that sends the requests to the classification component for processing. In addition to predictions of appliances running, total power, voltage, and current values for every second between a start and end time are available from the API. These values can also be downsampled arbitrarily using the API.

The front facing web dashboard and PR site are created using Django, a python based web app framework. This takes care of user details and associations between users and their SEADS hardware. The dashboard shows a line graph of current energy usage, as well as a bar graph of daily energy usage for the past week. The time series data is presented in a graph that allows the user to click and drag over a date range to label a period of time with a set of appliances that were running during that period. When the user does this, the label is sent to the server to use for training the models. The graphs are implemented using a library built on top of d3.js called C3. C3 provides an interface that abstracts away the minutia of d3.js to easily create visually appealing representations of data. Cross origin requests to the prediction REST API is handled at the client layer. This allows the dashboard to directly request data from the server using the prediction REST API, speeding up loading and response time.

Conclusion

SEADS-P allows for prediction and classification of power data. It also stores historical power usage in our database. The technologies assembled in our application create a scalable, extensible architecture that lends itself to improvements of the predictive algorithms we use. As the system grows it could be adapted to disaggregate energy loads of large office buildings or other commercial structures.

Dealing with power fluctuations is a vital part of the Intelligent Energy Management System (IEMS), but SEADS-P focuses on optimizing electricity usage in individual houses. Installing the SEADS device to monitor power usage has the potential to save considerable amounts of money and energy in the long term. In the short term it helps balance a grid that is becoming increasingly powered by intermittent renewable energy sources.

IEMS already handles optimization on the macro scale, but efficiency on the micro scale has yet to be tackled in a convincing way. If everyone could efficiently control their power usage in their homes, we could save millions of dollars in electricity every year while effectively combating the inherent fluctuations in power due to renewable energy sources. Eliminating waste is a great way to help the environment as well.

SEADS-P is an innovative and game-changing way to integrate renewable energy into our daily lives, and is part of the path to a clean and sustainable future for all of humanity.

Acknowledgments

Ali Adabi, Patrick Mantey, the **Center for Information Technology Research in the Interest of Society (CITRIS)**
CENSEPS, **UCSOLAR**, and **NSF** for supporting this project. This work is a capstone project for the Senior Design courses under the supervision of Professor Linda Werner.