

Business & Technology Surveillance

Automated Load Forecasts for Better Demand Management

By Kevin McElwee, President, Brown Analytics and Consultant to NRECA

APRIL 2022

Business & Technology Surveillance

Automated Load Forecasts for Better Demand Management

By [Kevin McElwee](#), President, Brown Analytics and Consultant to NRECA

APRIL 2022

SUBJECT MATTER EXPERT ON THIS TOPIC

David Pinney

Program Manager— Analytics, NRECA Business and Technology Strategies, David.Pinney@nreca.coop

This article is a product of the [Analytics, Resiliency and Reliability Workgroup](#).

ARTICLE SNAPSHOT

WHAT HAS CHANGED?

Energy storage deployments continue to rise at cooperatives as hardware prices fall. Reducing peak demand, also known as *peak shaving*, is one of the most popular uses for energy storage and has the potential to significantly reduce demand costs for cooperative members. To fully realize the potential of energy storage, an accurate forecast of future load is needed to determine the best time to charge and discharge the storage.

WHAT IS THE IMPACT ON ELECTRIC COOPERATIVES?

NRECA has worked with its members to develop a load forecasting application, “CoopForecast.” It imports load and weather data and automatically generates day ahead load forecasts using machine learning methods. Burt County Public Power District in Nebraska has deployed the application to predict its day-ahead load for insights into how best to control its 6 MWh storage system.

WHAT DO COOPERATIVES NEED TO KNOW/DO ABOUT IT?

Consider what your strategy is for load forecasting, and please reach out to David Pinney, NRECA Analytics Program Manager at David.Pinney@nreca.coop, if you are interested in deploying CoopForecast.

Introduction

The NRECA team has created a simple-to-use, automated, day-ahead load forecasting web application that allows grid engineers to make better informed storage dispatch decisions for peak shaving. We have recently deployed the application with Burt County Public Power District (BurtCo), our first user. This article explains the advantages and technical details of automating load forecasting for co-ops.

Acknowledgement

Thank you to technical lead Zach Olson at BurtCo and lead business analyst Hasan Rammaha at Landis + Gyr for their help and feedback in implementing this web app at BurtCo. Hasan Rammaha was instrumental in partnering with BurtCo and helping to automate BurtCo's load data ingestion.

Overview of the Problem

A utility may decide to invest in batteries, direct load control, or behavioral and incentive-based demand response programs to help shave peak demand; however, these efforts are only as effective as the utility's ability to predict the day's energy consumption. The smallest inaccuracy can mean the difference between tens of thousands of dollars — implementing a peak-shaving strategy with incorrect load predictions can even increase demand cost.

Thankfully, advances in deep learning and neural networks can offer utilities an accurate picture of the next day's energy consumption. Building off work started by the Open Modeling Framework ([OMF.coop](https://openmodelingframework.com)), NRECA is releasing CoopForecast, a web application that generates a day-ahead load forecast for purposes of demand management. It automatically pulls in historical demand data, past weather behavior, and multi-day-ahead weather forecasts.

CoopForecast uses neural networks to create a day-ahead load forecasting model that can be easily implemented to inform dispatch decisions.

For demonstration purposes, we have provided a "static site" (coopforecast.com/demo), meaning that no data can be uploaded and all buttons will not be functional. If you are

interested in deploying this website with your own utility's data, please contact David Pinney, Analytics Program Manager: David.Pinney@nreca.coop. There is no cost to use the application, deploy your own instance, or use the source code at <https://github.com/dpinney/coopforecast>.

CoopForecast is built with the Python libraries Flask and Tensorflow. It is built to be as hands-off as possible. Even if a user has no expertise in data science, the site should show a clear forecast of the next 24 hours of load without the user having to modify any data.

Experience With Deployment at Burt County PPD

The first user of CoopForecast was Burt County Public Power (BurtCo). BurtCo in partnership with Bluestem Energy Solutions, installed a 1.4 MW solar array and a 6 MWh battery. It is one of the first in the Midwest to feature the Tesla Megapack. The project is also the very first solar and utility-scale battery storage system to be built in tandem in Nebraska.

When asked about their experience with the CoopForecast, BurtCo technical lead Zach Olson responded:

"Like most utilities, we get billed a demand charge on system peak. Previous to working with NRECA and Brown Analytics, BurtCo was estimating what the load would do for the day. Now, with the load forecaster, it will help us identify when to discharge the battery at peak times. The forecaster gives us another tool to use to make sure that we are getting the greatest value out of the batteries."

Model Generation Process

Every morning, hourly load data is posted to the site. A deep learning model is trained on this new data and forecasts the next 24 hours of load. The web app automatically syncs weather data provided by ASOS and the National Weather Service every hour. Then, a new model begins training. This is all automated, but users can also upload data and launch new models manually through the user interface, if setting up an automated data feed is too difficult.

For BurtCo, load data is published to the site at 6:30 a.m. every day. Forecast and historical weather data are automatically synced at 7 a.m., and then at 7:05 a.m. every day, the model begins to train on the days. A forecast is available to grid engineers shortly after.

Site Structure

The site is entirely password protected with data encrypted in transit via TLS. Other than the login page, no content is available without

a username and password. After logging in through the login page, the user can see the main site.

Forecast: After logging in, the user is immediately sent to see the latest successful forecast (see Figure 1). In light blue, it shows the last few days' load, and in dark blue, it shows the forecasted load for the next few hours. This page also includes information concerning accuracy and historical performance.

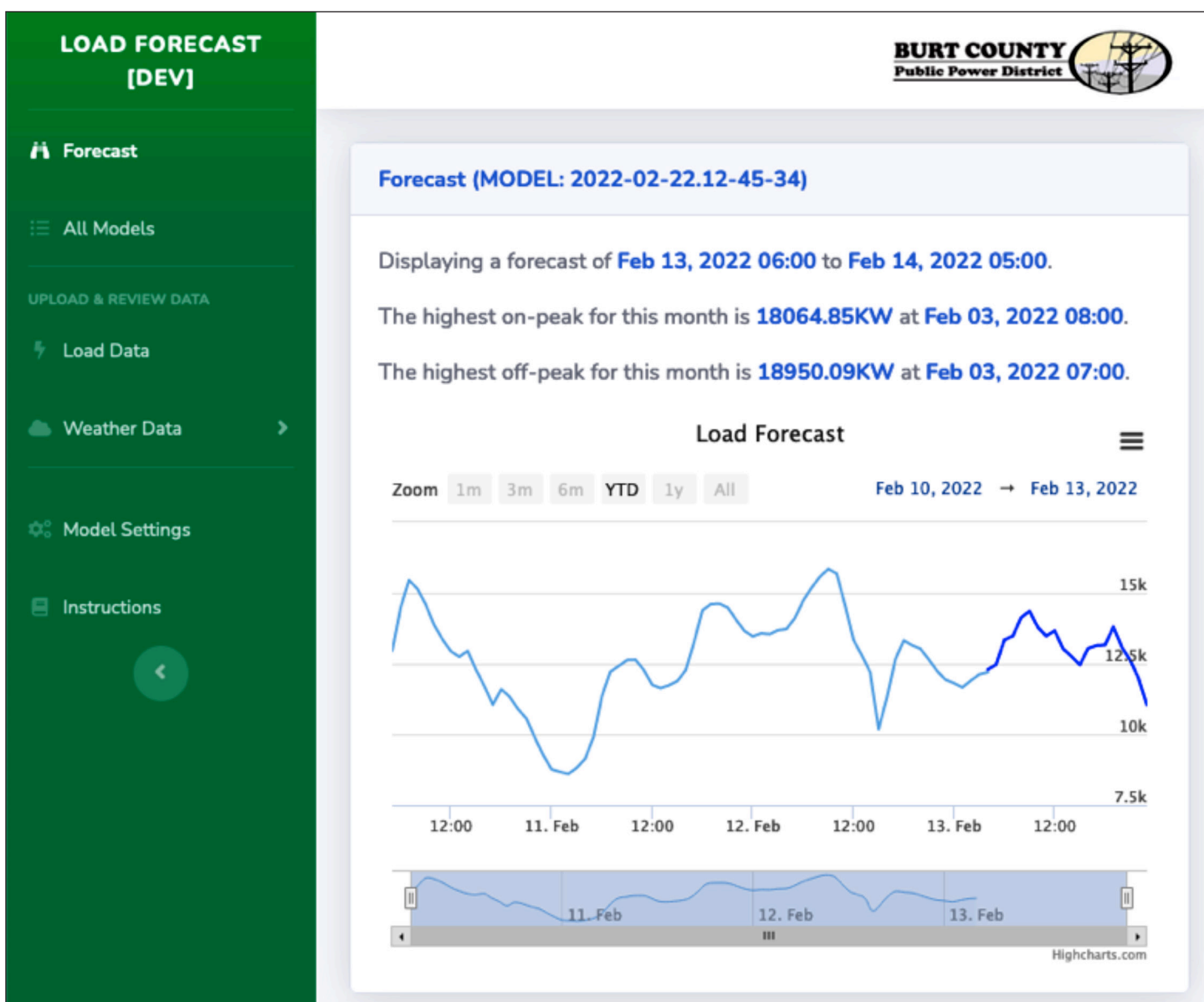


FIGURE 1: Forecast Page

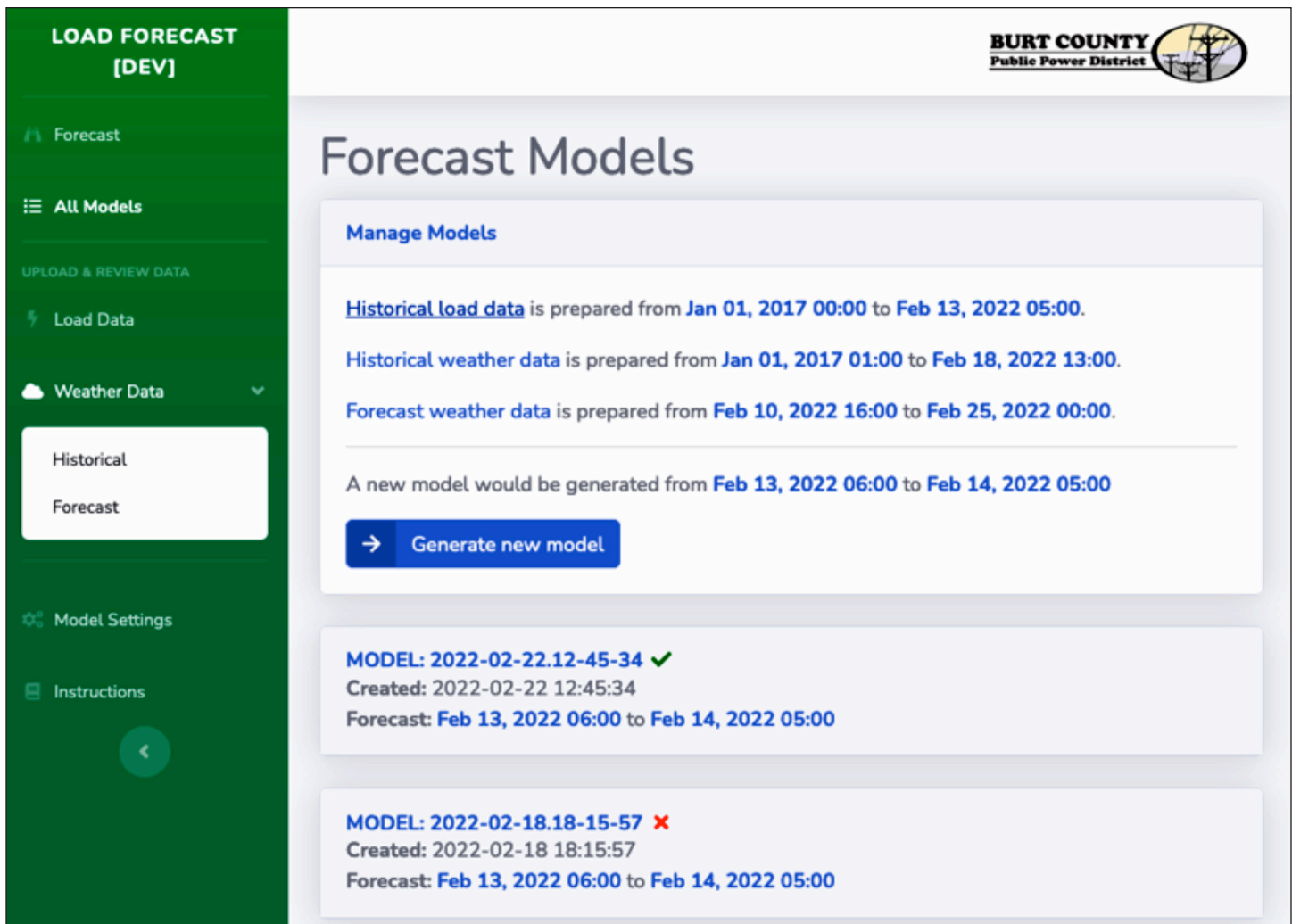


FIGURE 2: Models Display

All Models: CoopForecast displays all the models that the site has created. It also has a window that shows what data has been uploaded and their timestamps. If the user wants to generate a new model, the user simply clicks on the button that says “Generate new model” (shown in blue in Figure 2).

Data Views: There are three types of data that the site contains to make forecasts: Historical Load Data, Historical Weather Data, and Forecast Weather Data. Each has its own page where a user can upload a comma-separated values file (CSV) with timestamps. A summary, graph, and table of the uploaded data are available (see Figure 3).

Import information about missing data is also displayed. These values will be automatically filled in by the web app when the model is trained; however, in general, it is best to remove any long swaths of missing data as much as possible.

Model Settings/Instructions/User Settings: Various documentation and configuration options are available to users.

Demonstration site: For those interested in exploring the site further, a static version of the site is published at: coopforecast.com/demo. To be clear, a static site has no database or functionality. All buttons are disabled and no files can be uploaded. Data is provided by ERCOT.

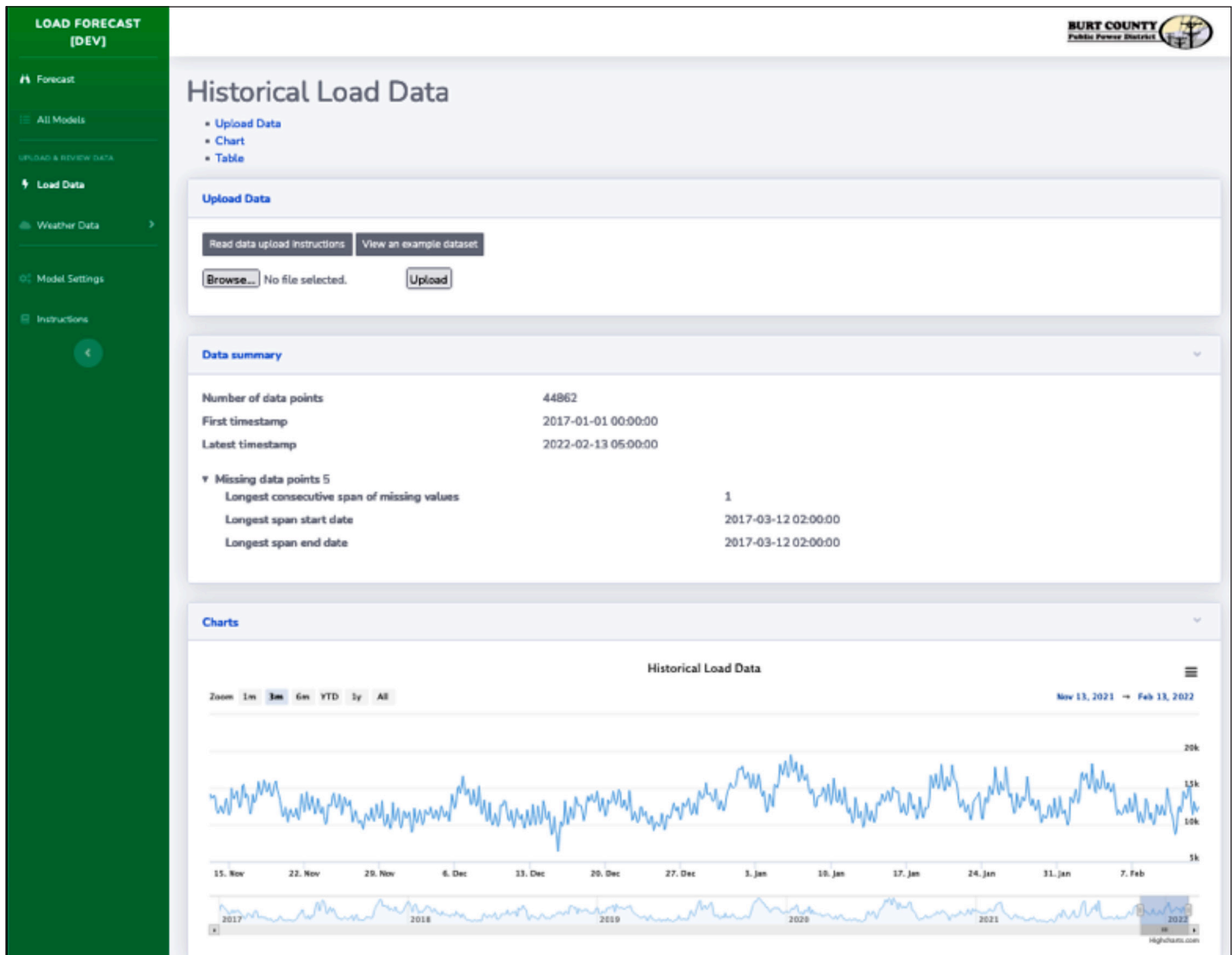


FIGURE 3: Historical Load Data Page Showing the Summary and Graph of the Uploaded Data (the Table is not Shown).

Software and Deployment

All the code for the site is open source and released under the [GNU General Public License v3.0](#). View the repository on GitHub: <https://github.com/dpinney/coopforecast>.

The site was deployed on an EC2 instance. It was built with Flask and Tensorflow. More details concerning the deployment process are documented in the repository's README. A single, idempotent installation script for deploying on a EC2 instance is included. All automated jobs are configured with cron, and all services are handled via systemd. A sqlite database was sufficient.

The site is built with Bootstrap's free theme [SB Admin 2](#). Charts are created via Highstock JS v9. Otherwise, there is minimal Javascript.

A collection of common commands is available via the [CLI](#), including syncing weather data, posting a file to an endpoint, initiating a backup, and deploying the site. All cron jobs are deployed using this CLI.

For local development, a Dockerfile is provided to set up the site locally; however, the site should be simple enough where Docker is not required, especially for Linux and Mac users. The site roughly follows a **"Model/View" design pattern**. Issues are tracked via GitHub; bug reports and PRs are welcome.

Utility-specific code (i.e., unique dispatch decisions, custom file parsing) was separated into a separate file: 'burtcoppd.py', in the hope that future utilities could easily separate out

functionality that did not align with the structure of their systems.

Data is formatted with [black](#), and test coverage is [published by codecov](#).

Automated Data Ingest

For a few years, the OMF has made deep learning technology accessible to co-ops, but CoopForecast saves grid engineers time by automating the data ingestion process from start to finish: no CSV uploads; no collecting and cleaning weather data. A grid engineer simply needs to log onto the site, and they will see a forecast.

HISTORICAL LOAD DATA

After logging into the site, a programmer can make a “POST request,” using a script to publish a file with load data to the site on a daily basis.

A formal API has not been created yet, but the script is no more than 15 lines of code and is documented in the CLI under the 'post-data' command. Essentially, a programmer creates a session, makes a POST request to login, and then using the same session, makes a second POST request to the specified endpoint to upload the data. Most programmers should be familiar with how to implement this in their language of choice.

With respect to uploading load data, there are a few things to remember:

- The load data should not contain shaved peaks. If you give shaved peaks to the model, it will harm the forecast accuracy.
- Publish the past 48-72 hours of load data or publish multiple times every day. If you publish the past 24 hours of load data every day, you are creating a fragile system with a single point of failure. If one morning the POST request fails, you will have to go in by hand to clean the data.

HISTORICAL & FORECAST WEATHER DATA

After getting load data uploads on an automatic schedule, utilities are effectively done: historical and forecasted weather data ingestion is entirely hands-free. Zipcode and time-zone are configured when setting up the site, which pulls in weather data from the National Weather Service and ASOS.

Further, if for some reason the hourly cron job is not working, the user can simply click “Sync with the National Weather Service” or “Sync with ASOS,” and that can also fill all the weather data required.

Deep Learning

Deep learning is certainly a complicated subject, but the fundamental operation of these models is straightforward: we are using correlation to make a forecast. Essentially, if given load and weather data from the past, how can we predict tomorrow's load?

For those interested in the details of this model, you are welcome to [read the code tied to BurtCo's specific model](#). In less specific terms, yesterday's load, day of week, month, year, and temperature are all used to inform the forecast. [Figure 4](#) is a diagram of how the model works.

For an in-depth overview of the machine learning model, please see our three-part article on [neural network models for load forecasting and peak prediction](#).

Further, since the site is modular and well-tested, data scientists who are interested are welcome to construct their own neural nets. The car is built; you are welcome to drop in your own engine.

To make sure the system is automated, a programmer can send a POST request to the site to create a new model. The code to implement this is available in the CLI under the command “launch-new-model.”

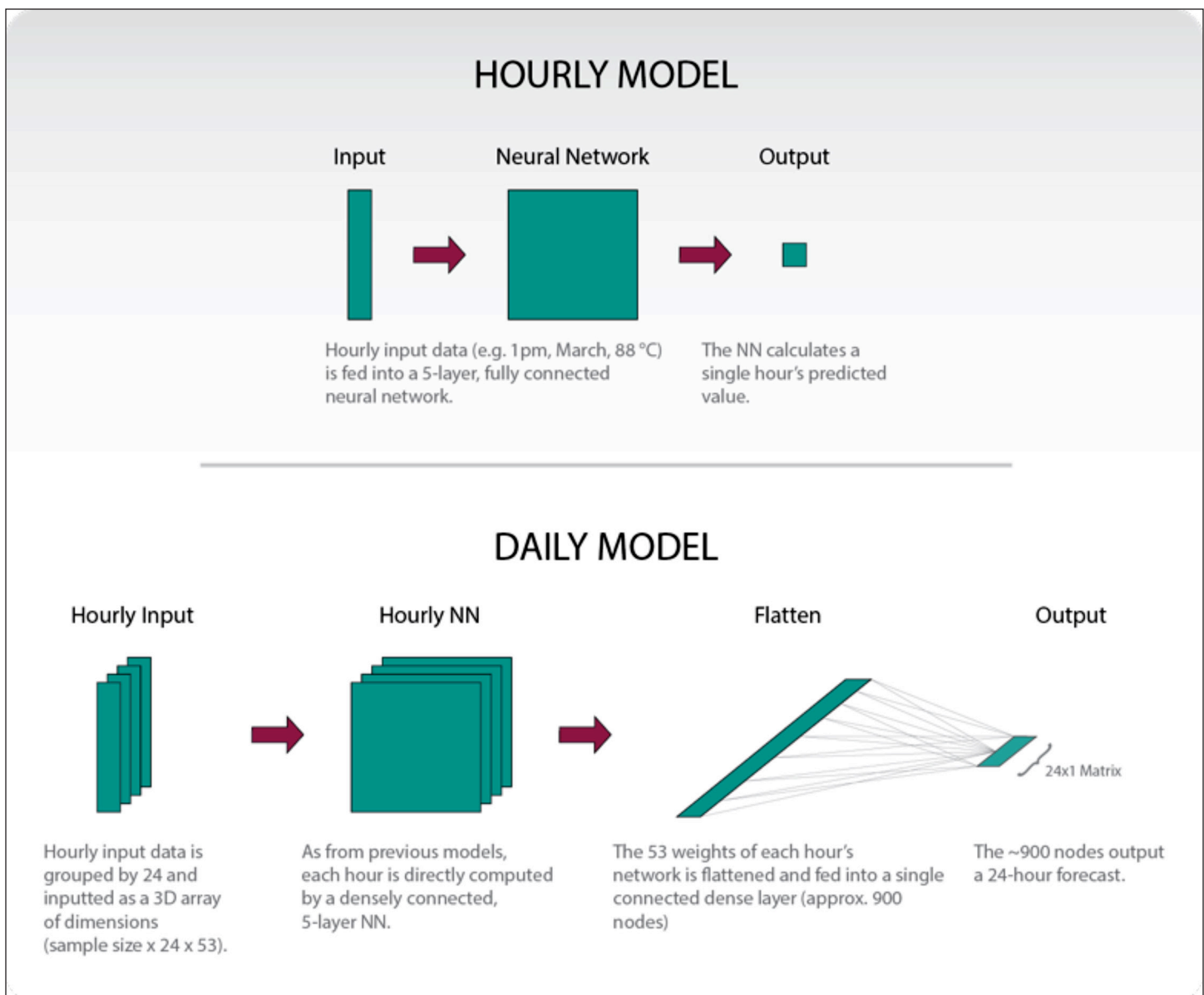


FIGURE 4: Using Correlation to Make a Forecast

Implementation for Other Utilities

The app has been in deployment at BurtCo for about a month, and BurtCo has confirmed that it consistently adds value to their operations. The model has performed at about 4 MAPE (mean absolute percent error) on the BurtCo load and local weather forecast data.

The site was built so that it could easily be re-deployed by any utility. In summary, the steps are:

1. Change Flask app's configurations to set the zip code, timezone, domain, password, etc.
2. Use the deployment scripts in the code's repository to set up a new version of the site at a new domain.
3. Upload historical load data and setup an automated feed.

No neural net structure will be universally perfect for every utility, so it is recommended that for each new installation, slight tweaks be made to 'forecast.py' for the best results. Assuming the utility has quality data and setting up an automated feed of daily load information is not too difficult, it should take very little time for a programmer to get this running.

Future Work

There are some limitations that users should be aware of when using CoopForecast. These issues may be addressed in future development.

- The site has only one login username and password.
- Daylight Savings Time is not observed. Users will only get a 24-hour forecast, and the model reduces all 25-hour days and increases all 23-hour days to 24-hour days. Forecasts on the two days of the year when Daylight Savings adjustments are happening should be carefully assessed against other forecast methods to guard against forecast error.
- There is a lot of potential for **daily dispatch recommendations and analysis** that can be added to the site. But every utility's storage and solar situation is different, and we have tried to make a tool that does one thing well, generalizable to any co-op. We hope that the software is modular and well-documented enough that someone from the NRECA team or a grid engineer can fork the repo and create custom analysis plugins as each custom need arises.
- We are planning to conduct a literature review and talk to vendors to better understand how the accuracy (MAPE) of the model compares to other tools that are available. ■

OTHER RESOURCES

- Supporting work from OMF.coop on **load forecasting and dispatch decision models**.
- An overview of the **development process for the OMF.coop forecasting technology**.
- For a discussion of how rates interact with demand, please see the **NRECA/CFC Retail Rate Guide**.

FEEDBACK WELCOME

NRECA welcomes thoughts, advice, and questions about CoopForecast or other areas of co-op interest. Please contact David Pinney at David.Pinney@nreca.coop.

ABOUT THE AUTHOR

Kevin McElwee is the president of the data science firm Brown Analytics. Brown Analytics develops analytical software for academic and non-profit clients with an emphasis on applied mathematics, machine learning, and interactive data visualization. Kevin's background is in software engineering, and he is a widely published data journalist and a researcher in the field of digital humanities.

QUESTIONS OR COMMENTS

- David Pinney, Program Manager – Analytics, NRECA Business and Technology Strategies, David.Pinney@nreca.coop
- To find more resources on business and technology issues for cooperatives, visit our [website](#).

ANALYTICS, RESILIENCY AND RELIABILITY WORKGROUP

The Analytics, Resiliency and Reliability (ARR) Work Group, part of NRECA's Business and Technology Strategies department, is focused on on current and future data and research required to provide prompt technical and economic support to the NRECA membership. Specifically focused toward the electric co-op community, ARR products and services include: development and maintenance of a portfolio of energy analytics products and services; collection and analysis of data; and provision of additional products and services in the areas of the data collection, IT architecture, sensors, and energy markets. For more information, please visit www.cooperative.com, and for the current work by the Business and Technology Strategies department of NRECA, please see our [Portfolio](#).

LEGAL NOTICE

This work contains findings that are general in nature. Readers are reminded to perform due diligence in applying these findings to their specific needs, as it is not possible for NRECA to have sufficient understanding of any specific situation to ensure applicability of the findings in all cases. The information in this work is not a recommendation, model, or standard for all electric cooperatives. Electric cooperatives are: (1) independent entities; (2) governed by independent boards of directors; and (3) affected by different member, financial, legal, political, policy, operational, and other considerations. For these reasons, electric cooperatives make independent decisions and investments based upon their individual needs, desires, and constraints. Neither the authors nor NRECA assume liability for how readers may use, interpret, or apply the information, analysis, templates, and guidance herein or with respect to the use of, or damages resulting from the use of, any information, apparatus, method, or process contained herein. In addition, the authors and NRECA make no warranty or representation that the use of these contents does not infringe on privately held rights. This work product constitutes the intellectual property of NRECA and its suppliers, and as such, it must be used in accordance with the NRECA copyright policy. Copyright © 2022 by the National Rural Electric Cooperative Association.