Final Technical Report (FTR)

Cover Page

Federal Agency	*Department of Energy*	
Award Number	DE-EE0009358	
Project Title	*Improving Grid Awareness by Empowering Utilities with Machine Learning and Artificial Intelligence*	
Project Period	*Start:* May 1, 2021,	*End:* January 31, 2023
Principal Investigator (PI)	Name: Cody Smith Title: Chief Technology Officer Email Address: cody@camus.energy *Phone Number: (415) 322-9127*	
Business Contact (BC)	*Name: Raj Raheja* *Title: Chief Financial Officer* *Email Address: raj@camus.energy* *Phone Number: (415) 322-9127*	

DocuSigned by:
Rajen Raheja
AFBC11640551450...

Signature of Certifying Official

7/1/2024

Date

By signing this report, I certify to the best of my knowledge and belief that the report is true, complete, and accurate. I am aware that any false, fictitious, or fraudulent information, misrepresentations, half-truths, or the omission of any material fact, may subject me to criminal, civil or administrative penalties for fraud, false statements, false claims or otherwise. (U.S. Code Title 18, Section 1001, Section 287 and Title 31, Sections 3729-3730). I further understand and agree that the information contained in this report are material to Federal agency's funding decisions and I have any ongoing responsibility to promptly update the report within the time frames stated in the terms and conditions of the above referenced Award, to ensure that my responses remain accurate and complete.

Acknowledgement: "This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) Solar Energy Technologies Office (SETO) under the DE-FOA-0002243, Award Number DE-EE0009358."

Executive Summary

Existing grid data sources and monitoring methods are generally stovepiped into operational data (e.g., SCADA), customer or billing data (metering), and 3rd party data (forecasting) repositories. Further, the value of this data is limited by gaps, errors, and a lack of data fusion. This segmentation reduces the value of the data by limiting real-time, market, and planning analyses for solar PV.

Camus Energy has created a cloud-based situational awareness software platform for understanding grid behavior in dynamic environments with high levels of distributed energy resources (DERs). Working with researchers at Pacific Northwest National Laboratory (PNNL), we empirically tested, downselected, and incorporated advanced machine learning (ML) and data analytics (DA) methods into our collection and analysis pipelines with the aim to apply ML/DA methods to two primary data sources: loads/injections at the network endpoints and flows and voltages over the network.

Using ML/DA methods integrated into the Camus pipeline, we were able to create unified network endpoint time series data across data inputs by detecting and correcting gaps, estimating time series data at unmetered network endpoints, and providing intraday and day-ahead forecasts for all network endpoints.

The activities/research tasks were organized into five major areas: 1) Use case and data management, 2) Endpoint Data Processing and Analysis, 3) Network Models and Situational Awareness, 4) Software integration, and 5) Integrated Software Performance Verification. The technical scope also included:

- Developing high-quality, unified historical, real-time, and forecast time series data for all metered and unmetered network endpoints (customer load and PV generation and larger solar PV generation sites) and exogenous environmental variables (i.e., solar insolation) that drive the behavior of network endpoints.
- Developing a physics-informed methodology for network data correlation which is robust to inaccurate and incomplete data sources, even in the absence of a power system model.

Through this project, Camus Energy was able to generate state-of-the-art grid analytics tools by improving upon existing open source tooling already available, and adding our own open source code for available public use.

This report describes findings from three endeavors: (1) End-Point Data Analysis; (2) Intra-day Solar Forecasts; and (3) Reduced Order Network Models (ROM).

Table of Contents

List of Figures

List of Tables

1 Background

Gap filling time series data typically depends on linear interpolation [1]. More recently gap filling advancements include machine learning techniques [2]. However, none leverage advanced learning approach that uses cohort training [3] or a neighborhood informed approach, which is described in this report.

The report also describes a physics informed approach using Reduced Order Models (ROM). There are several methods to capture the nature of the detailed system in aggregated models, however there is a trade-off for these methods developed for multiple applications [4]- [5]. These methods have specific requirements and applications that includes consideration of dynamics or covering a larger range of operating conditions, etc. The various methods of aggregation are:

1) Thevenin equivalents for downstream networks [4], [6]
2) Equivalent feeder representation to capture downstream network losses accurately [7]
3) Structured reduced order models for dynamics [8]
4) System identification-based ROM (abstract dynamical model) [5]

Methods described in items 1 and 2 above are ideal for steady-state models and useful for this application. Of these two methods, based on the data availability, the targeted application, the reduced order model that is proposed to be developed is the equivalent feeder model representation. This includes a structure of the reduced order model whose parameters can be determined by the system load and losses with the meter measurements.

2 Project Objectives

Clean energy goals are threatened by a lack of understanding of the true impacts of distributed energy resources (DERs) and their real-time impact on grid operations. Solar developers in particular have identified delays in the interconnection process as a key barrier to both utility-scale and behind the meter DERs. Secretary Granholm has repeatedly identified hosting capacity and impacts analysis as a key challenge in deploying sufficient renewable energy to meet our clean energy deployment and climate targets. For both small rural cooperative utilities and large investor-owned utilities, gaps and lags in real-time data, stovepiping of data, and high penetration of unmetered PV all make quickly and accurately performing DER interconnection analysis nearly impossible.

Unfortunately, data reliability is a significant issue. In some cases, data might be missing from a particular meter for large periods of time. In other cases, the data may be sparse and have missing data scattered throughout the data set. This report addresses these issues by exploring both model- and machine learning (ML)-based methods for filling in missing data.

Through this grant, we were able to increase the availability and value of the situational awareness on the distribution grid through real-time gap filling, back casting, nowcasting, near casting, and forecasting throughout the system from net system load to feeder heads, to individual meter endpoints and photovoltaic (PV) production meters. We leveraged two techniques for applying machine learning algorithm (XGBoost) to fill in missing data:

1. A nowcasting approach that trains using cohort data sets and estimates missing metered data without training on the meter itself. The outputs from this model produced mean square error results that were comparable to common ARIMA and persistence-based approaches.
2. Using a meter's neighbors to estimate missing data values. In this case, the algorithm also used and produced results that generated high r-squared values.

In each of these approaches the net metering of PV creates unique challenges for the analysis. Therefore, the report explores and defines a PV forecasting approach. The outputs from the forecasting are useful for improving the two gap filling approaches. The PV forecasting implementation used HRRR to forecast the solar irradiance at defined locations. The irradiance was then provided to a PV model and estimated the power output during clear and cloudy conditions.

In some cases, the data could be so sparse that data-driven methods will not work and therefore require a physical model to represent the system well. In past literature, models were shown to represent systems with accuracy, but require significant time and effort to create, calibrate and maintain. With that in mind, this work implemented a reduced order model (ROM) that would require less work to operate. The simulation results of the model used to represent a single feeder in Northern New Mexico were

promising and could potentially provide utilities with the necessary information to make important decisions.

3 Project Results & Discussion

The work conducted under this award was broken out into 5 subtasks as described in the Statement of Project Objectives (SOPO). Task 1 (data, use-cases management, and advisory board) consisted of convening a technical advisory board to provide input on data and use cases; the results of this task are described in Section 3.1. Task 2 (customer load and short-term PV data) consisted of gathering, processing, and cleaning end point data, including gap filling; the outcome of this task is described in Section 3.2. Task 3 (Network models and situational awareness) consisted of developing a parameter estimation algorithm and comparing that to a power flow model; this work is described in Section 3.3. Task 4 (Software integration) focused on integrated parts of the modules from Task 2 and 3 into the Camus software platform and is described in Section 3.4. Finally, Task 5 (Integrated software performance verification) aimed to test the performance of the integrated software, as described in Section 3.5.

In addition to the outcome of the tasks described in section 3, Table 1 shows whether, when, and how the milestones described in the SOPO were achieved.

Table 1: Summary of Milestones Achieved

Milestone	Description	Achieved	How
1	The power flow model and the GridAPPS-D / GridLAB-D solver will yield a converged power flow model for > 95% of typical cases	Yes (July 2021)	Subtasks 3.1 and 3.2
2	Adaptation and assessment of quantitative measures of accuracy of the ML/DA methods used for network endpoint analysis	Yes (November 2021)	Subtasks 2.1
3	Availability of 6 months of historical PV reforecast data in the Camus software environment	Yes (March 2022)	Subtask 4.3
4	Prototype of advanced Ritta software completes an end-to-end analysis of endpoint and network data in < 10 sec for the selected subsection of the utility network with:		
4.1	Synchronized data with granularity of < 1 minutes for all data sources, including lower-frequency sources which rely on model-based estimates to enhance time resolution	Yes (July 2022)	Task 4 (machine learning model-based methods for fill, but at 60-minute intervals)
4.2	Model-based network-level situational awareness available at < 5-minute resolution	Yes (July 2022)	Task 4 (machine learning model-based methods for fill, but at 60-minute intervals)
5	The Camus software system achieves the following performance targets: End-to-end collection and processing		

	latency of < 10 sec for high-fidelity data sources		
5.1	Synchronized data (including un-metered PV) with hourly granularity available across the distribution system.	Yes (November 2022)	Task 4 (machine learning model-based methods for fill, but at 60 minute intervals)
5.2	Intra-hour forecasts for the network endpoints available at < 5 minute resolution	Yes (January 2023)	Task 4 (machine learning model-based methods for fill, but at 60 minute intervals)
5.3	Model-based network-level situational awareness available at < 5 minutes resolution	Yes (January 2023)	Task 4 (machine learning model-based methods for fill, but at 60 minute intervals)
5.4	Composite system model that integrates measured and interpolated data in < 1 sec	No	Did not run physics model of the system that integrates measured and interpolated data.
5.5	< 10% error in inferred load / generation for missing, dropped, or unmetered endpoints; < 1% error in voltage and aggregate power flows in the medium-voltage network	Yes (January 2023)	Task 5

3.1 Task 1: Data, Use-cases Management and Advisory Board

3.1.1 Data Management Plan

The project team consulted with Kit Carson Electric Cooperative (KCEC) to determine the data sets used, update the Data Management Plan, and provide a high-level document to summarize the use-cases as communicated by our utility partner.

The Camus Energy team worked closely with Kit Carson Electric Cooperative to understand their needs in understanding the impacts of high penetration, largely unmetered PV on their system. At the time, Kit Carson was working towards their goal of 100% daytime power sourced from solar (since achieved in December 2022). We chose the Arroyo-Hondo (A-H) feeder as the first system we would explore. The A-H feeder has high renewables penetration, a high concentration of smart meters, and what seemed to be a power flow model with recent updates.

Table 2: Data sources obtained on Arroyo-Hondo feeder

Data Source	Type--Resolution	Record length	Achieved
Transmission	1 minute	6 months	No transmission data collected
SCADA	5 minute	10 months	Yes, but at 15-minute interval

Customer AMI	15 minute	3-6 months	Yes
	Monthly energy	1-2 years	Not obtained
Large PV meters	Time series-- 1 minute	7 months	Yes
GIS network and asset data/models	Static	n/a	Yes

The selected feeder had approximately 3,000 meters, 60 distributed PV systems, and an average load greater than 4 MW.

The one-year of Kit Carson AMI data is available for use for given researchers or organizations obtain an NDA with Kit Carson for its use. Contact Richard Martinez, COO, Kit Carson Electric Cooperative, rmartinez@kitcarson.com, to execute an NDA[1].

In addition, sample AMI load data and weather data used to validate the forecast models described in sections 3.2.1 and 3.2.2 is available online without an NDA in a Google Cloud Storage Bucket. For the work described in Section 4.2, the PNNL power flow parameter estimation is also available at the same bucket.[2]. Readme files are included for both sets of data.

3.1.2 Use-Cases Management

The project team worked with the Technical Advisory Board (TAB) to develop a set of high-level use cases for situational awareness with the end goal of converting enhanced data streams into actionable information for distribution grid operators and planners. Discussions with the TAB centered around gaps between the ideal and actual data environment for electric distribution utilities. Figure 1 describes the ideal data environment where all endpoint time series data is captured at 1 minute (or better)

[1] Upon completion of the NDA, users may reach out to Camus Energy at seto-2243@camus.energyfor access to the non-public storage bucket: https://console.cloud.google.com/storage/browser/kcec_data;tab=objects?forceOnBucketsSortingFiltering=true&project=seto2243&prefix=&forceOnObjectsSortingFiltering=false

[2] Available freely with any Gmail account: https://console.cloud.google.com/storage/browser/seto2243-forecasting;tab=objects?forceOnBucketsSortingFiltering=true&project=seto2243&prefix=&forceOnObjectsSortingFiltering=false
Or via API without a Google identity here: https://storage.googleapis.com/seto2243-forecasting/
Further documentation on how to access publicly available Google storage buckets is here: https://cloud.google.com/storage/docs/public-datasets#how_to_use_public_datasets_on

resolution, there are no data gaps or errors, and all endpoint data is available in near real time (<10-30 seconds).

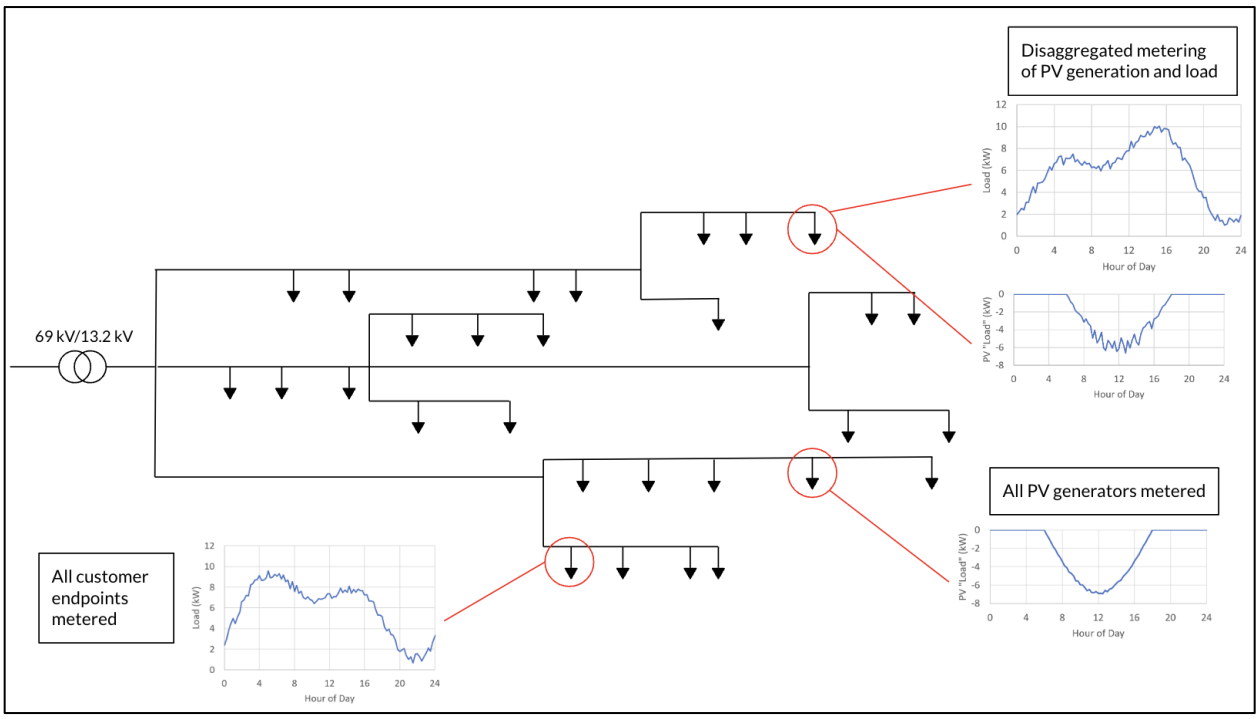


Figure 1: Ideal data environment for electric distribution utilities

Figure 2 describes the actual data environment for many electric distribution utilities where the resolution of data is limited to monthly readings, often with a delay, and has gaps and/or errors. Additionally, local generation, such as distributed PV, is not separately metered, meaning utilities do not have insight into gross demand and end points.

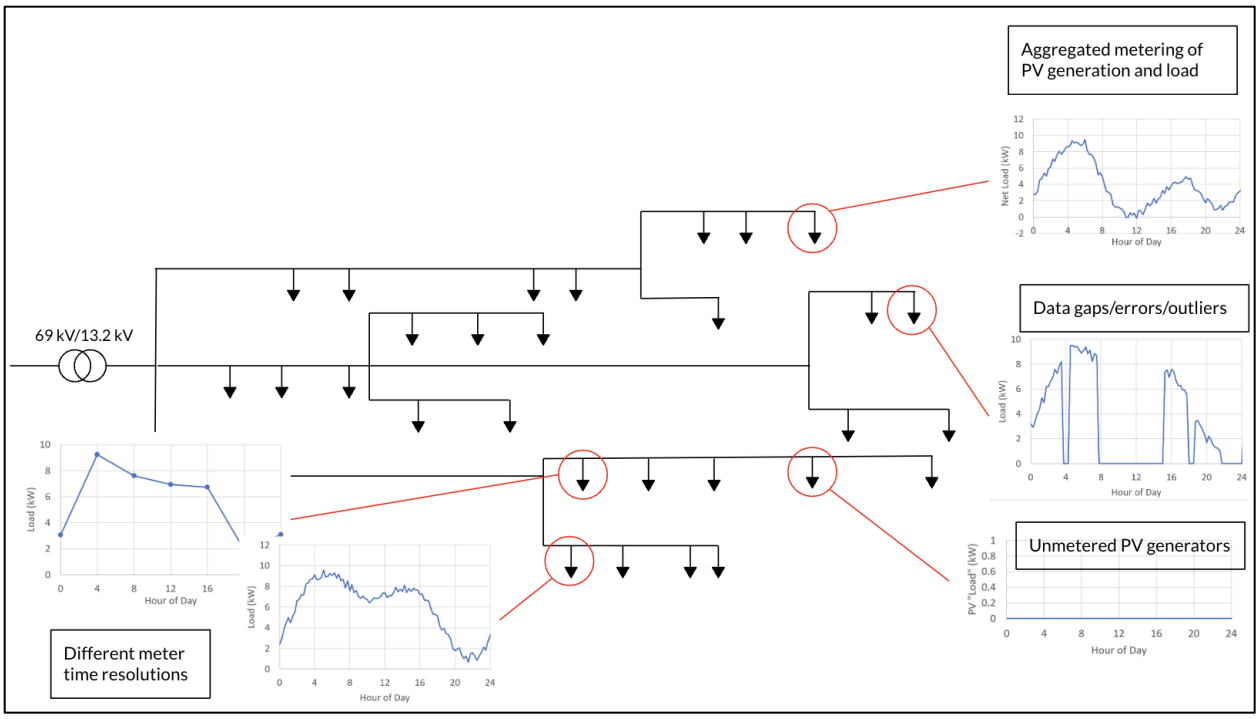


Figure 2: Actual data environment for electric distribution utilities

Based on this, the TAB focused on two use cases for an enhanced data environment:

1. Case 1: Input data includes individual meter
2. Case 2: Input data does NOT include the individual meter

Use Case 1 represents the approach that most power systems and operational engineers at a distribution utility take in situations of sparse meter endpoint data on a section of the grid with a growing number of behind-the-meter (BTM) DERs. This case assumes that power flows on upstream assets (nearer the substation) can be estimated by aggregating data in the downstream flows. This estimation/aggregation is more formally and accurately done by solving power flow equations for the network based on endpoint/meter loads. Solving the power flow equations gives you the voltages, as well. But given sparse meter endpoint data, the aggregation step requires both net and gross forecasting (and now-casting given the significant delays in rural distribution cooperatives' RF mesh networks to deliver advanced metering infrastructure back to the head end).

Use Case 2 addresses situations where hourly meters are sparse on the feeder, if they exist at all. There are many distribution cooperatives that have monthly (or more infrequent) reporting of usage at meter endpoints. With such sparse data, Use Case 1 is not really viable, and this is a technical fallback. The utility installs some sparse additional sensing on the trunks of circuits to provide data for power flow and voltage. These data would be used in statistical methods to infer flow and voltage downstream of the measurement points. Given more distribution utilities are adding hourly (or more

frequently) advanced metering infrastructure (AMI), we focused more on Use Case 1 for this project.

3.1.3 Technical Advisory Board

The Camus team recruited a Technical Advisory Board (TAB) to help ensure the impacts of this research were properly targeted to real life utility challenges. The TAB members are shown in the table below.

Table 3: Membership in Technical Advisory Board (TAB)

Name	Position	Organization
Bryan Hannigan	CEO	Holy Cross Energy (Glenwood Springs, CO)
Dan Harms	Executive VP of Grid Solutions and Special Projects	La Plata Electric Association (Durango, CO)
Soumya Kundu	Staff Research Engineer	PNNL
Emma Stewart	Chief Scientist	NRECA
Chris Campbell	Senior Director of Distribution & Telecom Operations	Salt River Project (Tempe, AZ)

The technical advisory board met 4 times throughout the duration of this project including once in person at the Distributech conference in Dallas, TX.

The purpose of convening the TAB was to agree upon a set of high-level use cases with the end goal of converting enhanced data streams into actionable information for distribution grid operators and planners. Specifically;

- For the TAB to provide feedback on the applicability of the results to other electric utilities, especially concerning the impacts of varying data quality from different utilities on the methods used in this project
- For Camus to report on interesting / useful findings as the project progressed, as well as challenges and difficulties in achieving the intended deliverables.
- For the TAB participants will provide feedback and recommendations on the status and progress of the project.

Specific questions posed to the TAB during these meetings were:

- Are we solving the right data problems?
- Are we planning for the right use of power system modeling within Camus's platform?
- What are use cases for ML-enabled now- or near-casting?
- How do we integrate research in user experiences?

The TAB facilitated use case definitions, as outlined in Section 3.1.2.

3.2 Task 2: End-Point Data Processing and Analysis

3.2.1 Customer Load and Short-Term PV Data

3.2.1.1 Endpoint Data Pre-Processing

Advanced Metering Infrastructure (AMI) data was provided by the utility partner: Kit Carson Electric Cooperative (KCEC) in Taos, New Mexico. The preliminary statistical analysis identified missing data within the yearlong set of AMI data. The cumulative distribution function (CDF) in Figure 3 shows how many data gaps occur for each length of gap, which is key information that guides our development and testing of ML/DA methods to fill these gaps. The assessment shows that 50% of the data gaps span 2 reporting periods (30 minutes) or less, and 90% of the data gaps span 10 reporting periods (150 minutes) or less. To fill the 30-minute gaps and remove 50% of the total number of gaps, we anticipate that simple interpolation will be sufficiently accurate. To fill the next 40% of the gaps and remove 90% of the total number of gaps, we anticipate that more sophisticated methods will be necessary. More sophisticated methods can capture and reproduce (quasi) periodic behaviors of the data towards more accurate interpolation than a simple straight-line approximation.

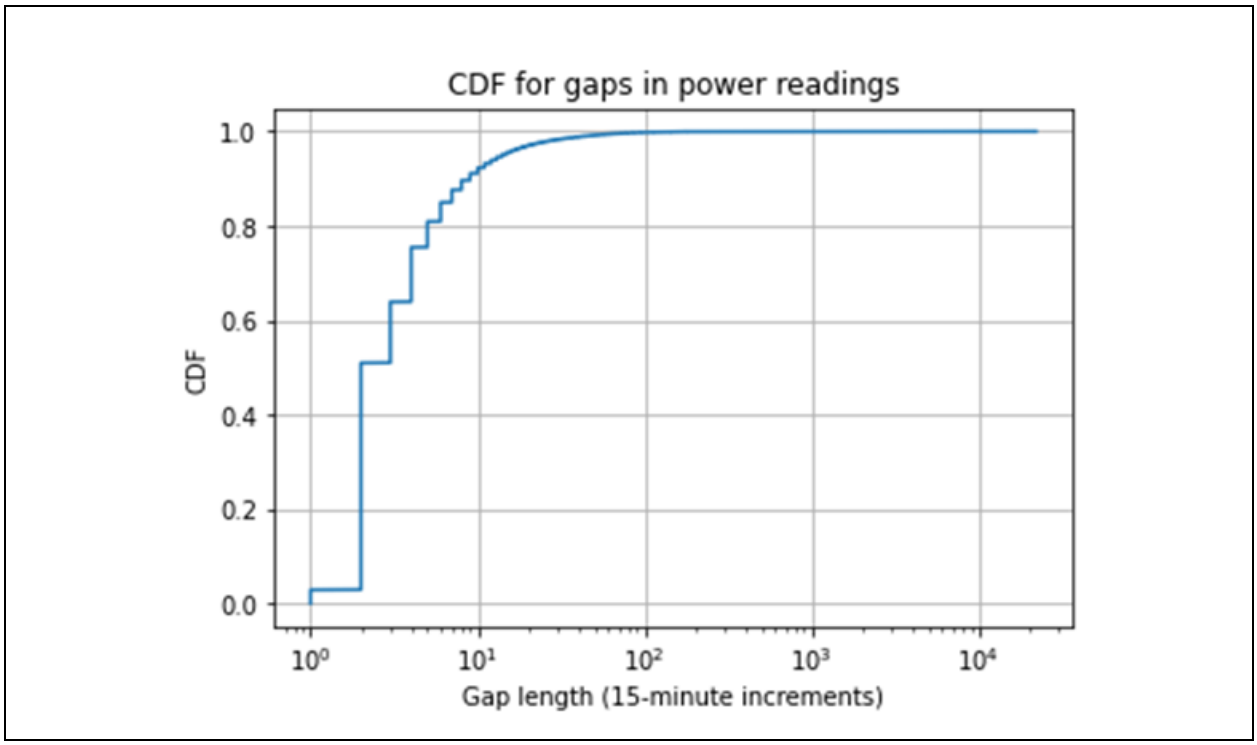


Figure 3: Cumulative distribution function (CDF) for the length of the gap in the KCEC AMI data. Each unit on the gap length axis represents a gap of one data reporting period or a 15-minute gap in the data.

Figure 2 shows a different interpretation of the impact of the gaps on the data received. Weighting each gap occurrence by the gap length gives a measure of data lost in that gap. Figure 4 shows the cumulative fraction of data lost for each gap length. In contrast to occurrence of a gap (in Figure 2), Figure 3 show that 60% of lost data would be recovered by filling gaps with a length of 10 reporting periods (150 minutes), and we will not reach 90% recovery of data until we fill gaps of 100 reporting periods (approximately 1 day in length).

The four subplots in Figure 5 show the probability distribution function (PDF) of the number of gaps of length N at each meter for N = 2, 4, 8, and 16, respectively. For N=2, nearly all the meters have between 2200 and 2800 gaps of length 2. This relatively narrow distribution shows that, for N=2, the failures that lead to these short gaps are spread relatively uniformly across the entire AMI fleet, i.e., there are no “problematic meters” that generate a large majority of these short data gaps. The remaining subplots in Figure 5 lead to the same conclusion for gaps of length N = 16.

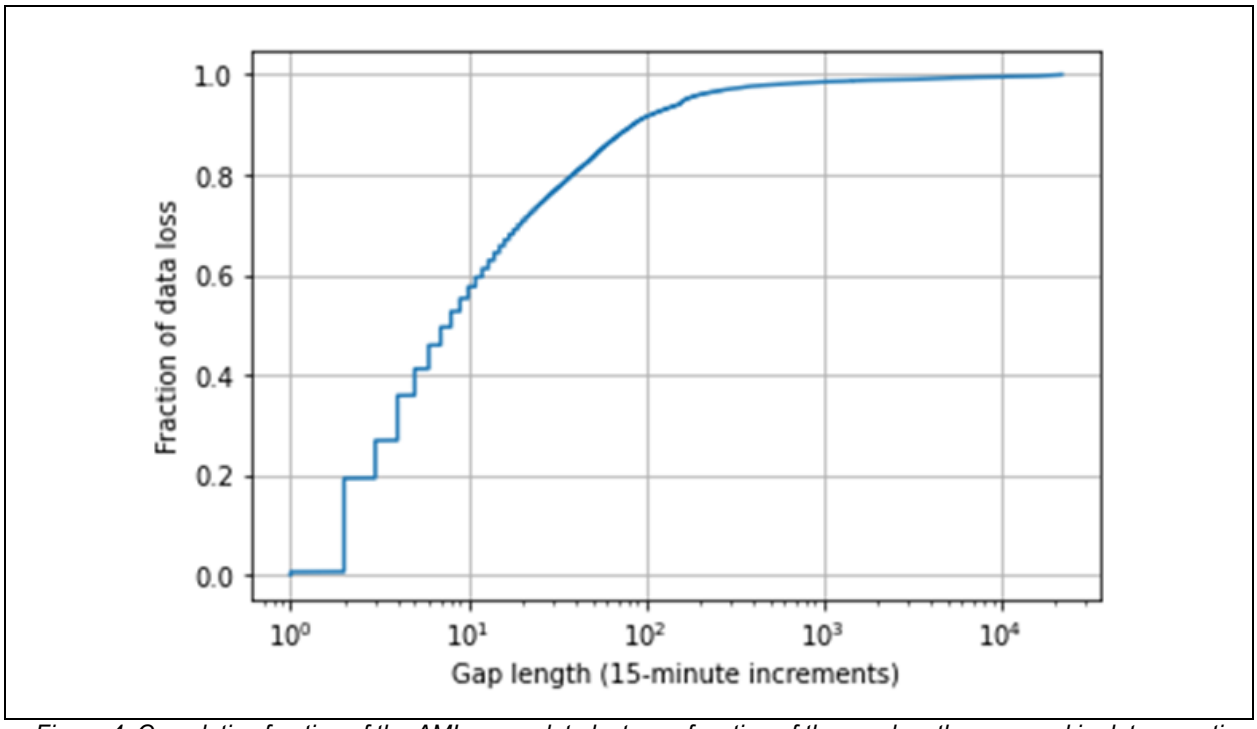


Figure 4: Cumulative fraction of the AMI power data lost as a function of the gap length measured in data reporting periods (one data reporting period = 15 minutes.)

3.2.1.2 Select ML Models for Gap Filling

Two data gap filling approaches were developed and tested using actual data from the field. Each used an XGBoost algorithm, but the training approach and inputs varied. One implementation took a cohort forecasting approach and is described in Section 3.2.1.3. A second implementation considered data from the neighbor’s power and voltage to estimate gaps (Section 3.2.1.3.2).

Extreme Gradient Boosting (XGBoost) is a supervised machine learning algorithm consisting of a distributed gradient-boosted decision tree. This approach has been proven useful for regression, classification, and ranking.

This algorithm uses decision tree ensembles, which include both classification and regression. The classification (often referred to as a decision) occurs as the tree is broken down into smaller and smaller subsets (or branches). The tree ensembles are useful in other modeling approaches, such as random forest. The gradient boosted tree approach differs in how training is administered.

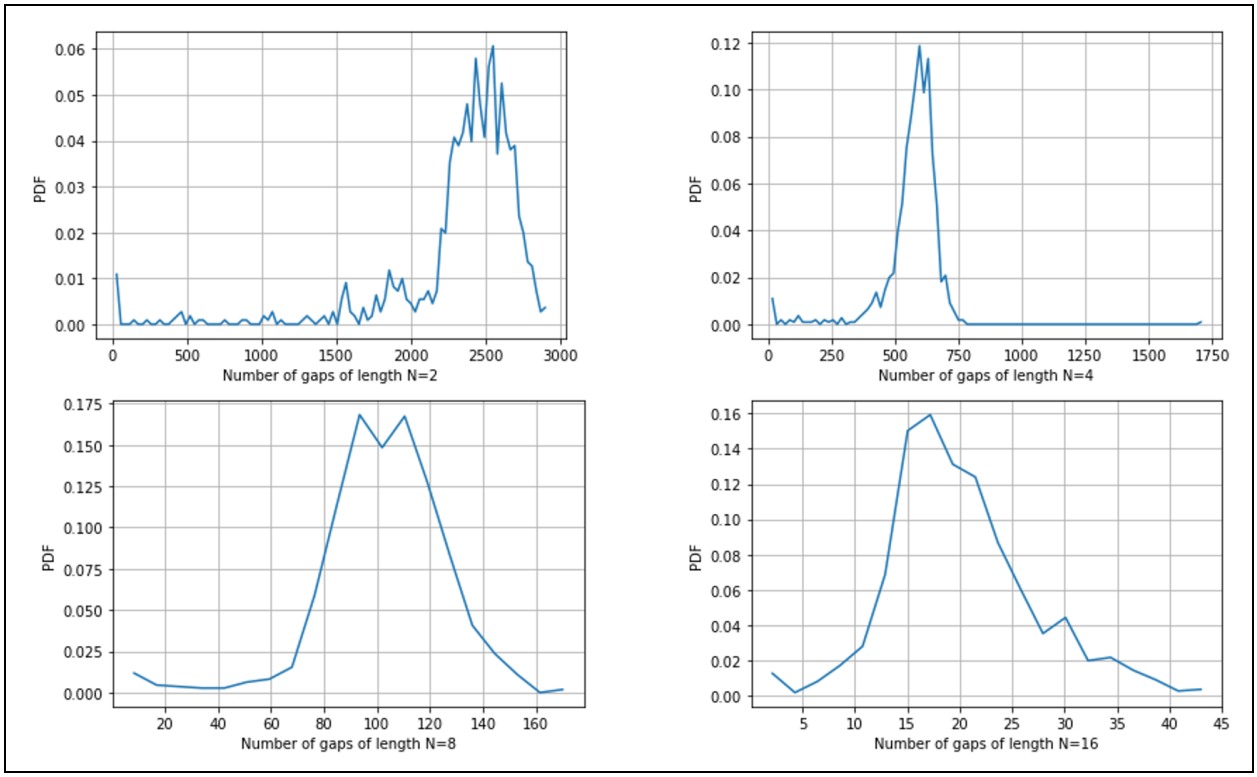


Figure 5: Probability distribution function of the number of gaps of length N at each meter for N = 2, 4, 8, and 16.

The objective of the training process is to find parameters (i.e., segments of the tree) that reduce the training loss. This is done through an additive training approach. This means that what has been learned is maintained and anything new is added upon observation of system states. The iterative training process estimates values that start off at zero at time step zero. Then at each step after zero a new estimate is made using the formed tree and the formation of the tree is created using an optimization algorithm. The prediction value (y) at each step (t) is estimating using the following:

$$\hat{y}_i^{(0)} = 0 \tag{1}$$

$$\hat{y}_i^{(1)} = f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \tag{2}$$

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \quad (3)$$

One key aspect of this training process is the model complexity, which in this case is referred to as the regularization term. Including the complexity slightly refines the tree definition to be:

$$f_t(x) = w_{q(x)}, w \in R^T, q: R^d \rightarrow \{1,2,\ldots,T\} \quad (4)$$

In the above equation, w is the leaf scores vector, q assigns the data point to each leaf, and T represents the number of leaves. The complexity w, is defined as:

$$w(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \quad (5)$$

3.2.1.2.1 Nowcast Estimates

Nowcasting is a portmanteau that blends the word now with forecast. This implies that forecasting prediction methods are used to estimate current operations. In this case, the forecasting model was the XGBoost supervised learning algorithm.

3.2.1.3 Implementation Method

The learning algorithm was deployed in a cohort environment where meters were grouped into categories. Training involved the exposure to a subset of data out of each of the cohorts. The learning model had no concept of the individual meter, nor of time. Which means that when training ends, the model can be used to estimate missing data for meters that were previously not part of the training set. The learning algorithm tries to find the internal parameterizations that minimize the error, but the model will not be perfect, and some errors will exist.

The nowcast models were trained using lagged features. The feature matrix includes back-shifted values, which are past values useful for predicting future values (i.e., autoregression). So, when testing occurred, to nowcast potentially unknown or missing values, past values of each meter were used as inputs.

3.2.1.3.1 Results

The implementation of the XGBoost algorithm, using the cohort training and testing approach, produced varied results. In most cases, the outputs accurately represented the system, as shown in Figure 6. But, in some cases, the outputs did not represent the performance of the meter well.

These results were compared with other common approaches including ARIMA and Persistence models. It turned out that using training with missing data points using the proposed approach resulted in root mean square error (RMSE) results that were greater

than the ARIMA and persistence results. The undesirable result is likely caused by the training that included gaps. Removal of the gaps is anticipated to improve the performance significantly.

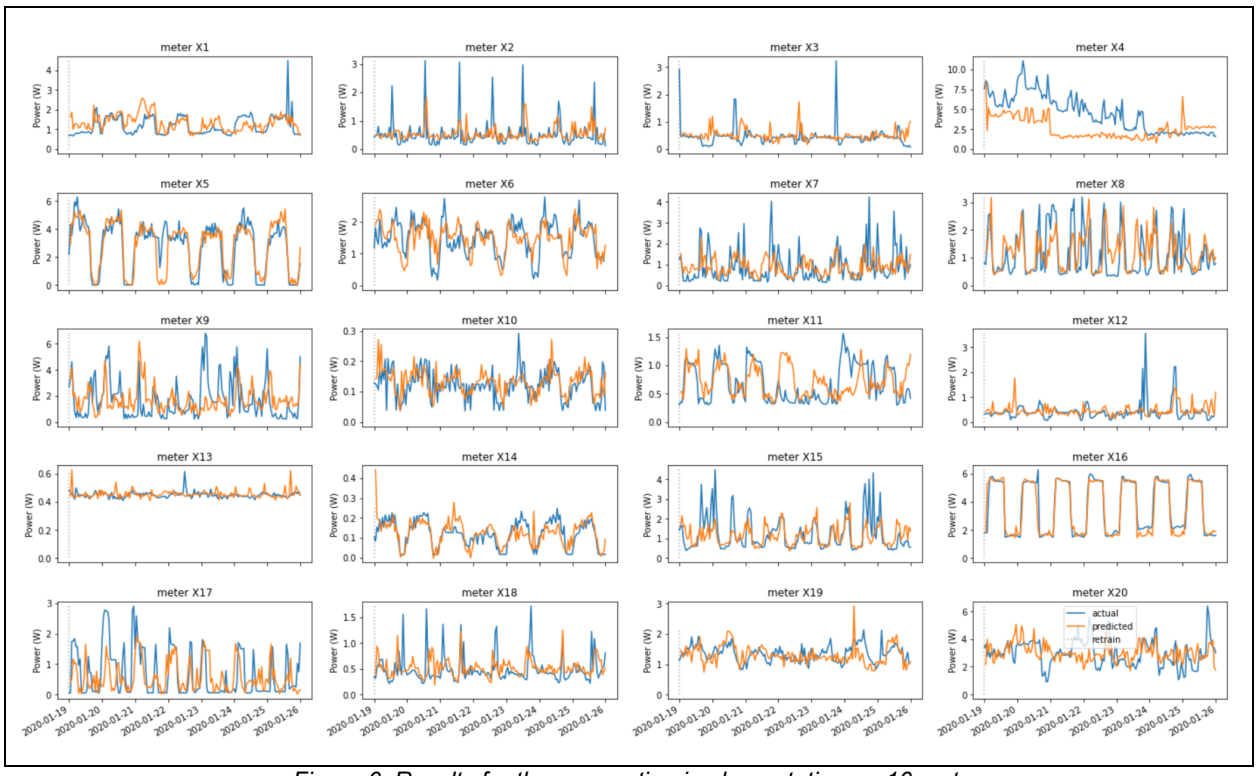


Figure 6: Results for the nowcasting implementation on 16 meters

The comparison of the three approaches, were tested on electric meters X1, X2, X3, and X4, shown in Table 4. As stated before, there was variation in the performance. For meters X1 and X2 the ARIMA and Persistence approaches did better in all cases. But, for meters X3 and X4, the ML was about the same as the 3-hr ARIMA and the 3-hr Persistence. The 6-hr ARIMA and Persistence did noticeably worse than the ML approach. On average, however, the ML did worse than each of the other four implementations.

3.2.1.3.2 Neighbor Informed Estimates

Likely missing data occurs at random meter locations and not in a concentrated area. Although not proven, an example of this idea is shown in Figure 7 where the number of missing data points for a single day are plotted as a heat map. Most of the meters with missing data in this example did not have any data for the entire day (i.e., 96 missing data points for an expected 15-minute data increment data set.) There were some meters that had a smaller number of missing data points, including 10 missing data points and 53 missing.

Table 4: Cohort model results comparison

RMSE (W)	**ML model (1 hr)** Grid search AR features only	**ARIMA (3 hr)**	**Persistence (3 hr)**	**ARIMA (6 hr)**	**Persistence (6 hr)**
Meter X1	0.397 (.451)	0.197 (.451)	0.199	0.230	0.265
Meter X2	0.181 (.118)	0.048 (.139)	0.052	0.054	0.059
Meter X3	1.14 (1.57)	1.132 (1.57)	1.191	1.476	1.361
Meter X4	1.40 (2.44)	1.451 (2.62)	1.042	1.635	1.492
Average of 4 meters	**0.865(1.14)**	**0.707 (1.19)**	**0.621**	**0.849**	**0.794**

Figure 7: Example map of missing data points for a day of operations

To estimate the missing data and fill in the gaps, the proposed method considered the measurements of nearby meters to inform a ML. More specifically, the data from nearby meters were used as inputs into an XGBoost algorithm [1] that outputs an estimate of the meters measured data. Using the neighbor's data to estimate gaps, especially gaps that extend more than 6 hours, will be more reliable than depending on its own lagged data to estimate missing points.

3.2.1.3.3 Community Analytics

Gap filling using the community data entailed a comparison of nearby meters with the meter that requires gap filling. Figure 8 depicts this method spatially. In the figure, the meter that requires gap filling is shown in blue and nearby meters are depicted with gray circles. A time series comparison of this evaluation is shown in Figure 9, where the meter (or node) that requires gap filling is in black, the average of the neighbors around it is in blue, and their standard deviation is indicated in the blue shade. The figure

includes the power and the voltage for both the meter in question and its neighboring group.

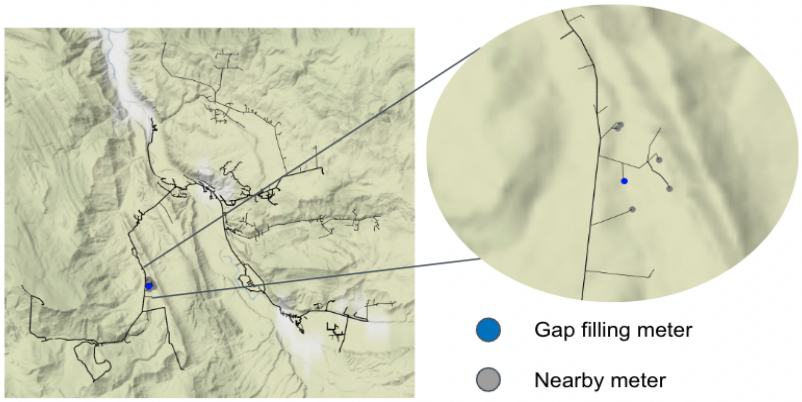


Figure 8: This image shows the spatial proximity of the gap filling meter with its neighbors

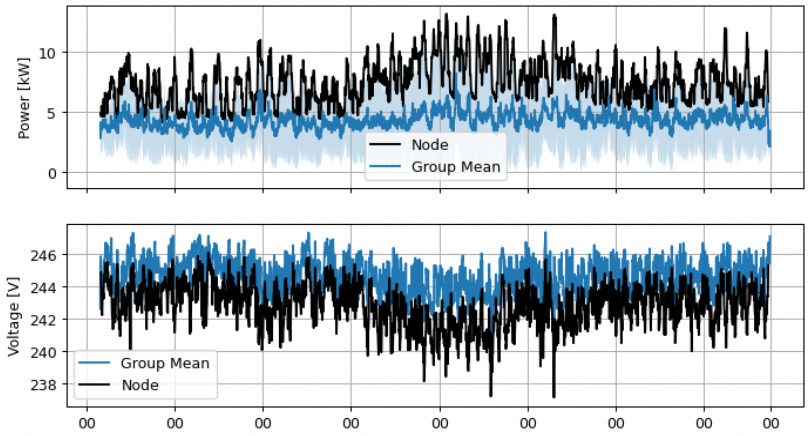


Figure 9: The time series power and voltage for the gap filling meter (node) and the neighboring meters statistics

The analysis tested two input approaches:

1. Input Option A:
    a. Group Average Power
    b. Group Standard Deviation
    c. Group Average Voltage
2. Input Option B:
    a. Group Average Power
    b. Group Standard Deviation
    c. Group Average Voltage
    d. Lagged value for the gap meter needing gap filling.

3.2.1.3.4 Gap Filling Results Using Inputs A

Using only the data from the group as inputs into the algorithm, the approach was able to represent the system relatively well. The top part of Figure 10 shows the training period in gray and the testing was where the red lines overlap with the gray. Over this

10-day period the estimate represented the general variation in the metered load will. It is however evident that the estimate included unnecessary noise that resulted in significant variation from the actual. This provides some evidence that the approach could be effective but requires further investigations of hundreds or thousands of meters to prove its overall accuracy.

3.2.1.3.5 Gap Filling Results Using Inputs B

The second implementation of the neighborhood-based approach included lagged values of the meter with missing data points. This would be effective in situations where the data is available to estimate the missing gaps. For this meter, the estimate followed closely with the actual values, as shown in Figure 10. In addition to capturing the general behaviors of the metered load, the extra input reduced the noise observed in the previous implementation that did not include a lagged value.

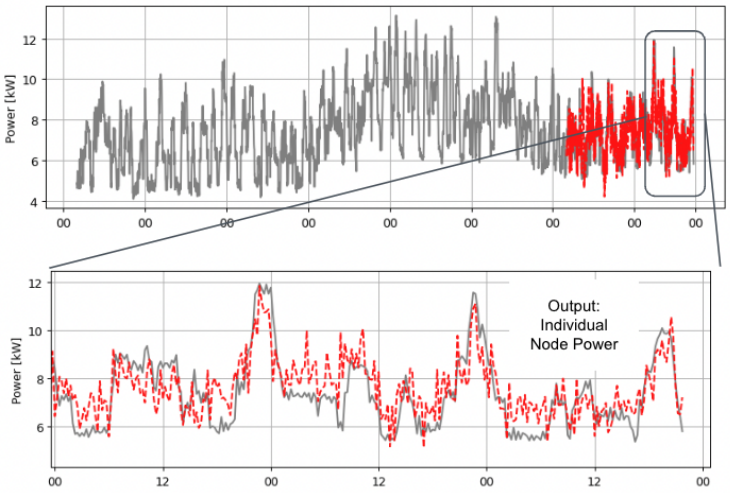


Figure 10: Sample results for the algorithm that used input option A for training

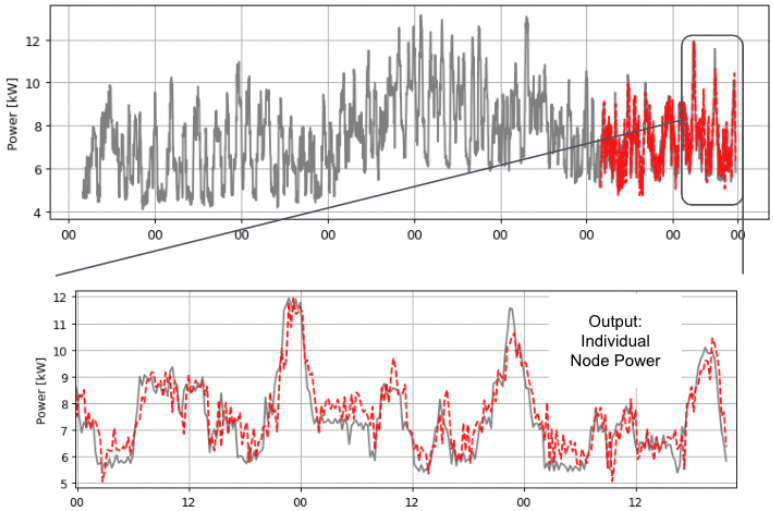


Figure 11: Sample results for the algorithm that used input option B for training

3.2.2 Intra-Day PV Forecast

3.2.3 PV System Modeling

Solar PV forecasts deployed in Camus's software uses the National Oceanic and Atmospheric Administration's (NOAA's) High-Resolution Rapid Refresh (HRRR) forecasts[3] and leverage's the National Renewable Energy Laboratory's PySAM model[4]. A summary of the solar PV pipeline is shown Figure 12.

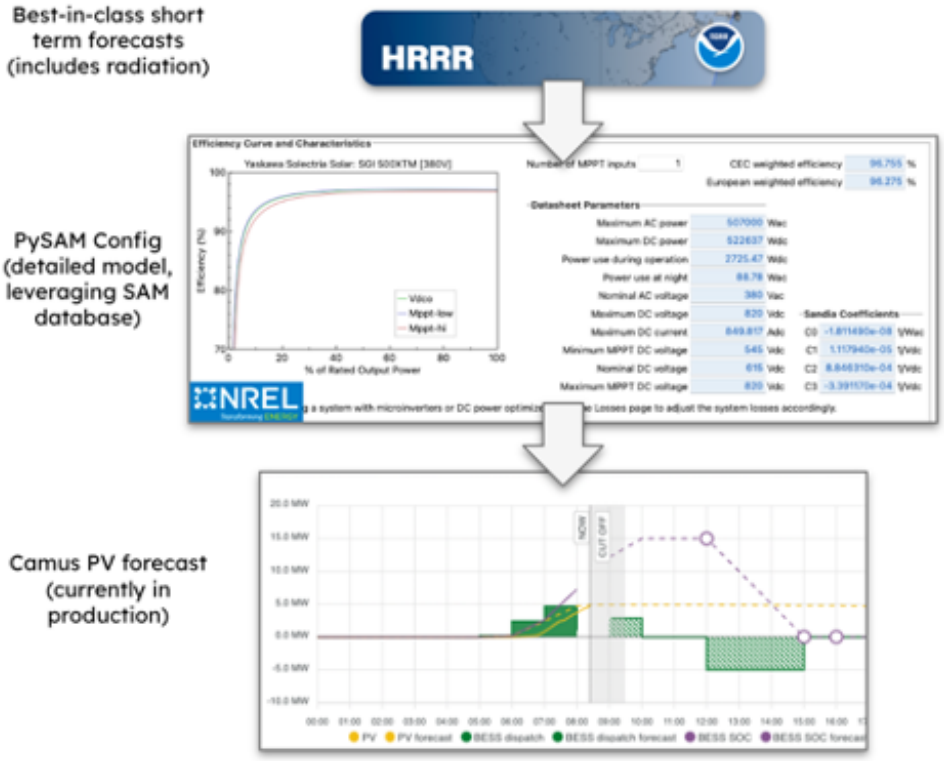


Figure 12: Solar PV forecast pipeline

HRRR is a cloud-resolving, convection-allowing atmospheric model, available in a 3 x 3 km grid, and are made every hour for up to 18 hours in the future, and every 6 hours for up to 48 hours in the future[5]. Data includes DNI, DHI, GHI, air temperature, wind speed, and visibility. Data is retrieved from a NOAA API endpoint in the grid file format, converted to zarr, aggregated by horizon, and then parsed for radiation data.

PySAM is a Python package that is used in Python code to make calls to SAM's simulation core, enabling access to many default values and component libraries. For smaller, customer-sited PV systems, where detailed information about the solar system (beyond rated capacity) is not readily available, we use the PVWatts implementation of PySAM, which requires only location and system size as inputs. Larger, utility owned and/or controlled systems are modeled in more detail to include specific inverter and

[3] https://rapidrefresh.noaa.gov/hrrr/

[4] https://sam.nrel.gov/software-development-kit-sdk/pysam.html

[5] Future work includes the integration of 15-minute forecasts as they become available from NOAA.

panel manufacturer, tracking, and orientation of the system. Current applications of solar PV forecasts (along with load forecasts) in Camus' software include informing the dispatch of large-scale storage systems, and accounting for the shadow load of un-metered DERs for grid restoration workflows.

3.2.3.1 PySam Model Validation

We used NREL's PySAM to model a 1.792 MW-DC/1.525 MW-AC solar PV system located at Sunnyside Ranch (SSR) in western Colorado. Key model inputs are described below:

- System size: 1.792 MW-DC/1.525 MW-AC
- Module: Hanwa Solar HSL (from spec sheet using the SAM simple efficiency module): Temperature coefficient: -0.41%/deg. C; Area: 1.995622 m2; Max power voltage: 36.8 Vdc, Open circuit voltage: 45 Vdc, Module structure: glass/cell/polymer - open rack, Module efficiency: 15.6 %
- Inverter: Yaskawa Solectrica SGI 500XTM (from SAM inverter library [2])
- Electrical configuration: estimated using SAM logic based on DC/AC ratio.
- Tracking and orientation: 1 axis tracking, 0-degree tile and azimuth, GCR: 0.3

To validate the SAM representation of SSR, we compared the 2020 observed system output with 2020 NSRDB data (2020 is the most recent year for which AMY data is available from the NSRDB). Because of the NSRDB's 4 x 4 km resolution, and the varied weather conditions in this location, we don't expect perfect alignment; results for 5 days in November are shown in Figure 13.

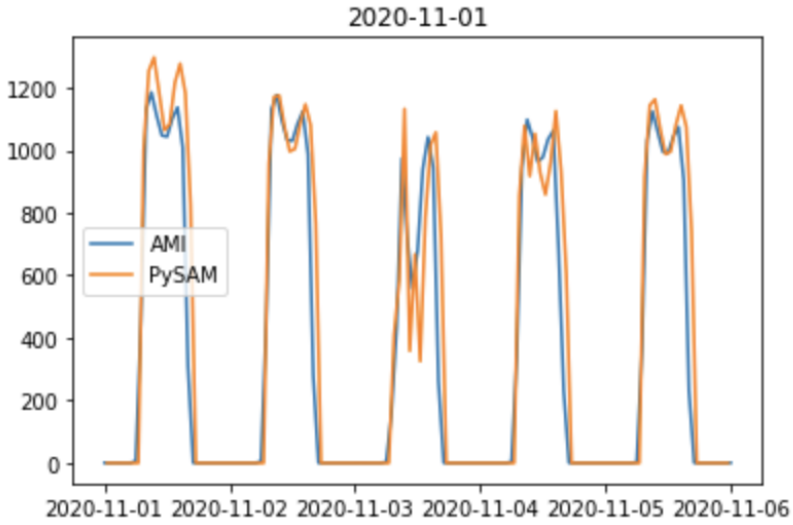


Figure 13: Comparison of actual versus modeled

3.2.3.2 Implementation

Finally, we re-forecast PV by executing the PySAM model as described above with NOAA forecasted weather. A comparison of PV forecasts with the actual AMI readings from SSR is shown in Figure 14 for 6- and 24-hour horizons. In general, the model shows reasonably good agreement with AMI data, both qualitatively and quantitatively. Note that for the 24-hour horizon case (right), a comparison is only possible every 6 hours when multi-day forecasts are made available by NOAA. Differences between the two are expected and are attributed to simplifications in model configurations (e.g.,

system architecture, inverter parameters, module variations), features not accounted for by the model (e.g., snowfall), forecasting error, and inaccuracies in AMI data.

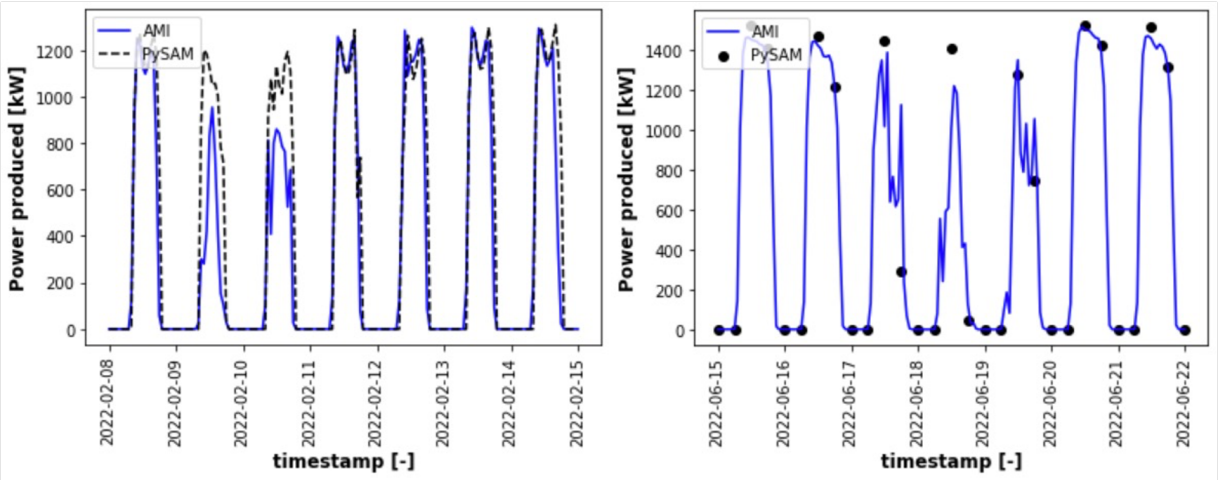


Figure 14: PV forecast for 6-hr horizon (left) and 24-hr horizon (right). The modeled results are shown in black and are compared with the actual data in blue

MAPE (Mean Absolute Percent Error) scores for forecasts made for various hours with several horizons (6, 12, 24 and 48 hours) over a six-month period- January 2022 to June 2022. The MAPE score for a given hour and horizon (say) hour = 11 and horizon = 6 represents the mean absolute error for PV power (kW) production at 11 am based on the forecast provided at 5 am local time. The MAPE for most hours is between 20-30%.

MAPE is not a perfect metric, because a few poor forecast values can skew the overall metric. This is clearly illustrated in Figure 15 where the computed MAPE for the three-day period is 34.77% even though 75% of the predictions have <7% error.

Hour of day	Horizon [hr] 0	6	12	24	48
9	28.15	25.88	25.58		
10	19.51	21.57	18.79		
11	32.89	23.61	24.35		
12	26.54	20.93	22.24	24.60	22.99
13	29.47	16.81	20.47		
14	33.14	21.65	32.89		
15	32.54	25.96	22.90		
16	27.12	28.52	25.70		
17	34.40	37.27	32.54		
18	37.85	41.44	33.77	46.17	43.92

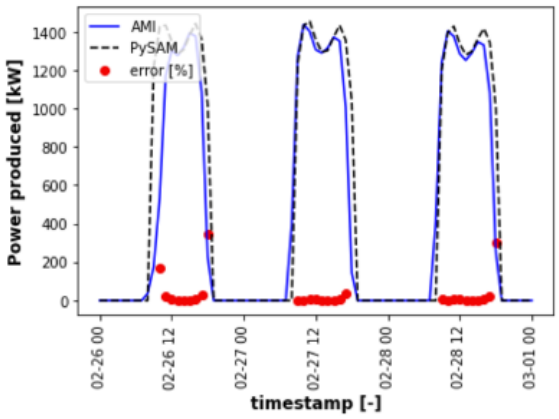


Figure 15: PV forecast for 6-hour horizon compared with AMI data

3.3 Task 3: Network Models & Situational Awareness

3.3.1 Physics Informed Methods

This work developed an equivalent feeder representation method. The method aggregated the down-stream distribution network into single loads and line losses. A detailed analysis used the IEEE test systems. Implementation defined the accuracies and the parameters sensitivities that impact the ROM parameter estimation.

The analysis showed that the ROM should be modeled as a ZIP load connected to an impedance to represent the losses correctly. The accuracy of the ROM is measured against the trunk node voltages observed in the complete model. The method accounts for a practical implementation by considering the available measurements: real power metering, voltage magnitude (no phasor information is available), no metering for trunk node voltages, etc.

After testing using the IEEE model, the Structured ROM (SROM) method was implemented on an actual utility feeder model. The analysis included AMI data and system models for Kit Carson Electric Cooperative (KCEC) in Northern New Mexico, U.S.A.

The Arroyo-Hondo (A-H) model in OpenDSS initially included many errors that were corrected to ensure the SROM implementation was accurate. Four trunk nodes were selected for aggregating the downstream parts of the system into a SROM. Synthetic AMI data was used to derive the SROM parameters. The accuracy of the voltages of the network with SROM and the full distribution system network were within the targeted ±0.005 p.u or ±0.5% of the absolute voltage values. Without recomputing the SROM parameters, based on the synthetic AMI data, the SROM power was updated for various operating conditions. The voltage errors for the varying operating conditions were also within the targeted worst-case errors of ±0.005 p.u or ±0.5% for the system model with SROM and full distribution system model.

3.3.1.1 Structure of the Reduced Order Model

The intent is to create a ROM that can capture the losses in the system with reasonably accurate results for slightly changing operating conditions with a similar load distribution in the system. The approach separates the load and the loss components of the network that is being aggregated. The simple method of lumping the losses along with the load does not capture the accurate behavior of the network losses for varying operating conditions. This method aggregates the positive sequence models for the three phase networks. The approach assumes the mutual coupling between the phases is minimal for the reduced order models, however the individual phase losses are translated into the ROM.

Figure 16 shows a single line diagram for a typical radial distribution feeder with zones in the feeder identified where the network needs to be reduced to enable meaningful system-level analysis. Figure 17 shows the radial distribution feeder with the ROM at the locations where ROMs are deployed to reduce the number of feeder nodes in the distribution network.

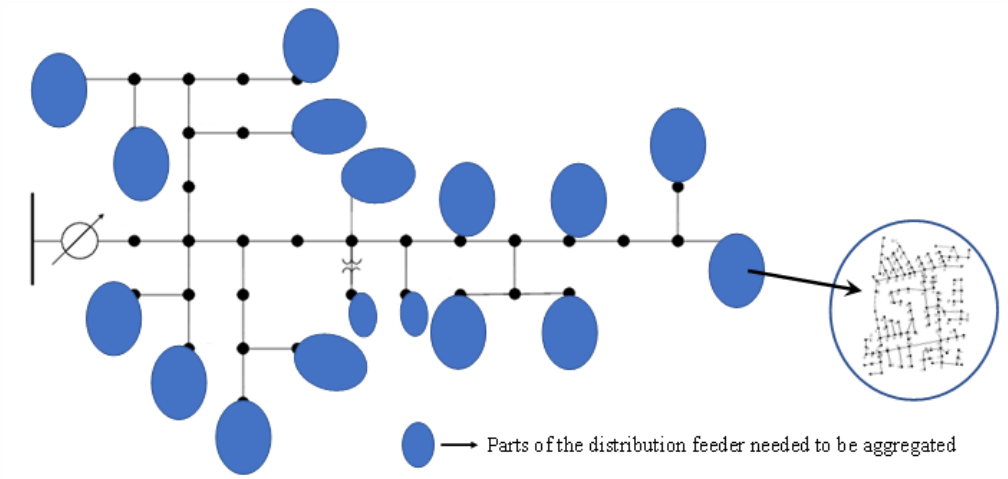


Figure 16: Single line diagram of an IEEE feeder with regions of the network that were reduced

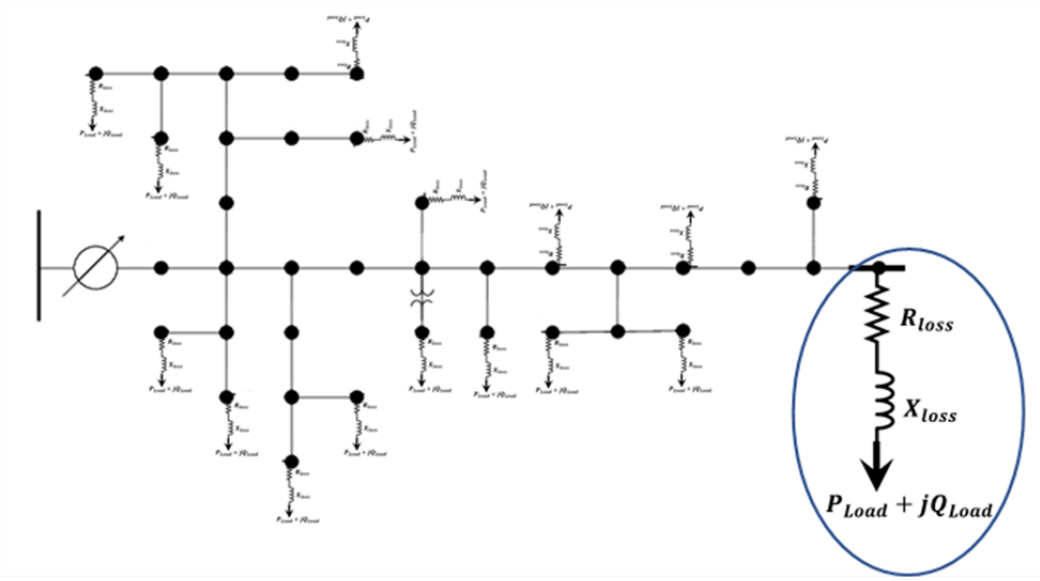


Figure 17: Single line diagram of distribution feeder with ROMs at the feeder terminals

3.3.1.2 Reduced Order Model Parameter Estimation

The approach assumes that the downstream network at the trunk node 'K' needs to be reduced using the above-described structure of ROM. Using simple power flow calculations and/or metering data the ROM parameters can be estimated by separating load and loss components of the distribution feeder. The equations below use complex powers (active and reactive power values). In real life the reactive powers may not be metered, however using the power factors, the reactive powers can be determined. For each phase, the below analysis is done to determine the three-phase ROMs.

$$S_{Total}^{K} = \sum_{i=1}^{N} S_{AMI} + S_{Loss} \; \forall \; Downstream\ nodes, i \; \in [1\,, N] \tag{6}$$

$$\left(P_{Load}^{K} + jQ_{Load}^{K}\right) = \sum_{i=1}^{N} S_{AMI} \;\; \forall \; Downstream\; nodes, i \;\in [1\,, N] \tag{7}$$

$$I_{Downstream}^{K} = \left(\frac{S_{Total}^{K}}{V^{K}}\right)^{*} \tag{8}$$

$$(R_{Loss}^{K} + jX_{Loss}^{K}) = \left(\frac{S_{Loss}}{I_{Downstream}^{K} \times I_{Downstream}^{K}{}^{*}}\right) \tag{9}$$

Where,

S Complex Powers
I Complex currents
V Complex Voltages
RLoss ROM Resistance that represents the network real losses.
XLoss ROM Reactance that represents the network reactive losses.

Equations (7) and (9) determine the ROM parameters from the metered data along with network analysis. The information available from metered data is real power, voltage magnitude at all nodes, power factors at all meter locations, network model to determine the losses and the complex voltages. The voltage magnitudes can be used to validate the network model simulations.

For the present project, we will develop methods to process the AMI data, utilize the network model solutions to determine the ROMs at the identified Trunk Nodes. We will utilize the information from the daughter nodes and where required metered data is missing, we will augment it with the network model solutions from distribution system solvers.

The algorithm that is developed to determine the ROM parameters based on the strategies highlighted above is given in Algorithm 1.

Algorithm 1:

1. Perform a detailed distribution system analysis on the model to estimate: (a) the average load power factor; (b) the average loss% for a region of the network that needs to be reduced; (c) record the powers resulting in a synthetic AMI meter data set.
2. Using the synthetic AMI data as the input, compute: (d) the expected losses based on the loss%; (e) the reactive power powers using the average power factor.
3. Using nominal voltage at the ROM location, determine the ROM equivalent impedance.
4. Adjust the ZIP profiles on the ROM for the three phases based on heuristics until the trunk node voltage error is <0.005 pu.
5. Validate the ROM using the field AMI data for different scenarios.
6. With the feeder impedance and load profile fixed, the ROM cam be represented with just the corresponding change in load.

3.3.1.3 Analysis and Results on IEEE 37-Bus Distribution System

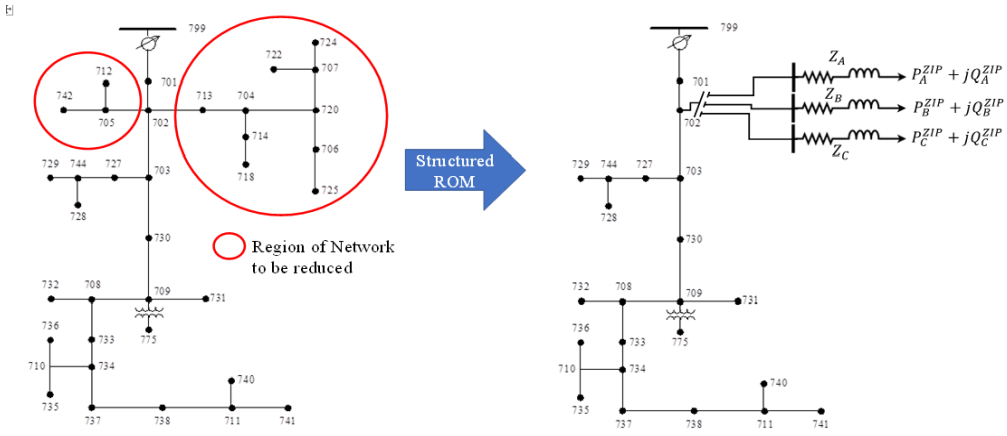
The IEEE 37-Bus distribution system was used to perform analysis on the IEEE test systems. The system has varying ZIP profiles which mimics real-world distribution feeder characteristics. Algorithm 1 described above, was used for the analysis and ROM development.

Figure 18 shows the IEEE 37-Node system with full network representation and the reduced system with the structured ROM. Figure 18 (a) shows the part of the network that is intended to be reduced in the red circle. The two parts are rooted to the trunk node 702. The IEEE 37 bus system is a 3-phase system that is not evident from the single line diagram, however the 3-phase structured reduced order model is shown in Figure 16 (b).

The accuracy of the structured ROM is determined based on the accuracy of the trunk node voltages for the model with the structured ROM (S-ROM) compared to that of the full model.

The trunk nodes where the voltage comparison is done are Nodes: 701, 702, 703, 709, 730, 775. The error is determined for voltage at each phase at these nodes as a percent of the deviation from the full system voltage:

$$V_{\phi_{error}} = \frac{V_{\phi}^{S-ROM} - V_{\phi}^{Full}}{V_{\phi}^{Full}} \times 100 \quad \% \qquad \phi \in \{a, b, c\} \tag{10}$$



(a) Full Network with Network to be Reduced *(b) Reduced Network with the Structured ROM*

Figure 18: IEEE 37-node feeder and network reduction using AMI data

3.3.1.4 Preliminary Analysis

Initially the ROM loads were set to be constant and based on the AMI maximum values. Full information from the model was used and assumptions of reactive power metering and realistic loss values were used. Under the unbalanced case, there was some missing loss data due to mutual coupling.

The system of equations was consistent and non-independent. This led to non-unique solutions, which resulted in the estimation of the mutual coupling impedance impossible in the S-ROM. However, using engineering judgement, the mutual impedance did not impact the load models.

Table 5 to Table 7 describe the model output accuracies at the trunk node using different methods for estimating the mutual impedances. There were three main parameters that determine the accuracy of the ROM:

1. The trunk node voltage used in Equation (3);
2. The Impedance computed in Equation (4), and
3. The Load profile (ZIP profile + power factor).

To understand these factors in detail and their impact, the project team evaluated several methods of deriving the S-ROM.

3.3.1.5 Deriving S-ROM parameters

Table 5 gives the errors of the voltages at the trunk node for each phase.

Table 5: SROM Accuracy w/ Measured Voltages

Trunk nodes	Va Error (%)	Vb Error (%)	Vc Error (%)
701	-0.34516	-0.2827	-0.61831
702	-0.62191	-0.49337	-1.07316
703	-0.63068	-0.49661	-1.07823
709	-0.63839	-0.49909	-1.08346
730	-0.63638	-0.49847	-1.08212
775	-0.64293	-0.49479	-1.08329

From Table 5 the errors on all the phases are quite high and this could impact the ROM accuracy significantly. The hypothesis was that the model could achieve an error of <0.5% (or about 0.005 pu).

Based on engineering judgment, the individual phase voltages completely decoupled the 3-phase S-ROM. An average voltage was used to compute the S-ROM parameters. Table 6 shows the accuracies with the average phase voltages.

Table 6: SROM Accuracy w/ Average Phase Voltage

Trunk nodes	Va Error (%)	Vb Error (%)	Vc Error (%)
701	-0.0867	-0.52758	-0.22577
702	-0.11692	-0.95559	-0.37774
703	-0.11989	-0.96012	-0.37908

709	-0.12234	-0.96307	-0.38054
730	-0.12169	-0.96234	-0.38016
775	-0.407	-0.65816	-0.40602

Table 6 data shows very high errors shifted from phase C to Phase B and phase A errors were computed to be much lower. However, the errors were still high. But the issue with this method was that the voltage measurements at the trunk nodes were not available in real life. Therefore, the nominal voltages at the trunk nodes were used. The results of this are shown in Table 7. The accuracy of the S-ROM derived from the nominal voltages match the accuracy with using the average voltages. This validates the approach that used the nominal voltages at the trunk nodes as the measurements.

Table 7: SROM Accuracy w/ Nominal Phase Voltage

Trunk nodes	Va Error (%)	Vb Error (%)	Vc Error (%)
701	-0.0867	-0.52758	-0.22577
702	-0.11692	-0.95559	-0.37774
703	-0.11989	-0.96012	-0.37908
709	-0.12234	-0.96307	-0.38054
730	-0.12169	-0.96234	-0.38016
775	-0.407	-0.65816	-0.40602

For the initial ROM method development constant P-Q loads were used to represent the ROM, and it clearly shows that this might be a drastic assumption as all the voltage errors with the S-ROM seem to be negative.

The team evaluated to use ZIP load profiles with the actual power factors from the reactive power measurements in each phase. ZIP loads can be defined as shown below:

$$P_{ZIP} = P_0\left(P_Z\left(\frac{V}{V_0}\right)^2 + P_I\left(\frac{V}{V_0}\right) + P_P\right) \quad (10)$$

$$Q_{ZIP} = Q_0\left(Q_Z\left(\frac{V}{V_0}\right)^2 + Q_I\left(\frac{V}{V_0}\right) + Q_P\right) \quad (11)$$

Where,
$P_0, Q_0 \rightarrow$ base real and reactive powers of the load
$P_Z, Q_Z \rightarrow$ constant impedance fraction of real & reactive power
$P_I, Q_I \rightarrow$ constant current fractions of real & reactive power
$P_P, Q_P \rightarrow$ constant power fractions of real & reactive power
$P_Z + P_I + P_P = Q_Z + Q_I + Q_P = 1$

An average ZIP profile of $[ZIP1] = [0.4\ 0.3\ 0.3]$ was chosen for the loads on all the three phases that indicates 40% constant impedance load, 30% constant current load and 30% constant power load. The accuracies with the ZIP profile were reasonable and are

shown in Table 8. The nominal phase voltages were used to determine the S-ROM parameters.

Table 8: SROM Accuracy w/ Nominal Phase Voltage and Avg. ZIP

Trunk nodes	Va Error (%)	Vb Error (%)	Vc Error (%)
701	0.070059	-0.20155	0.0492
702	0.147888	-0.37402	0.091086
703	0.149017	-0.37552	0.091893
709	0.150172	-0.37642	0.092621
730	0.149892	-0.37619	0.092443
775	-0.01919	-0.19116	0.07356

Table 8 errors also indicate that the ZIP profile could be different for different phases as the errors were consistently high and negative on Phase B. Therefore, the ZIP profile on Phase B was tuned and the corresponding accuracies are shown in Table 9. The ZIP profile for phases A and C were the average ZIP profiles ZIP Loads $ZIP_{AC} = [0.4\ \ 0.3\ \ 0.3]$ and the ZIP profile for phase B was tuned to get the errors low and was $ZIP_B = [0.9\ \ 0.1\ \ 0]$

Table 9: SROM Accuracy w/ tuned ZIP load profiles

Trunk nodes	Va Error (%)	Vb Error (%)	Vc Error (%)
701	0.006202	0.006306	0.051135
702	0.017011	0.016461	0.082249
703	0.017528	0.016556	0.082452
709	0.017814	0.016759	0.082675
730	0.017728	0.016708	0.08261
775	-0.01274	-0.00331	0.134358

This still has some challenges from practical implementation perspective. The loss and power factor metering. Considering both averaged power factors and losses, the S-ROM was tuned, and the parameters determined that resulted in accuracies shown in Table 10. The accuracies are very encouraging as can be seen from Table 10. The tuned ZIP profiles are $ZIP_{AC} = [0.4\ \ 0.3\ \ 0.3]\ and\ ZIP_B = [1\ \ 0\ \ 0]$.

Table 10: SROM Accuracy w/ tuned ZIP load profiles and average loss & load power factors

Trunk nodes	Va Error (%)	Vb Error (%)	Vc Error (%)
701	-0.04475	-0.01099	-0.00518
702	-0.07519	-0.01027	-0.01564
703	-0.07573	-0.01042	-0.01602
709	-0.07645	-0.0104	-0.01636
730	-0.07629	-0.01041	-0.01629
775	-0.09386	-0.05355	0.045776

3.3.1.6 Validation of the S-ROM for changing Operating Conditions

A few of the complete model loads were changed at random. Loads at nodes 712, 720 and 725 were changed that resulted in a reduction of 85 kW on phase C and 21 kW on phase B. For the S-ROM, The ZIP profiles, the impedances, the average power factor and average loss % were kept the same.

With just the new synthetic AMI real power data, the S-ROM load base power values were determined, and the accuracies were compared to the new full system voltages. The trunk node voltage errors are shown in Table 11. The overall accuracy of the S-ROM is reasonable and worst-case error within the targeted $\pm 0.5\ \%\ (0.005\ pu)$.

Table 11: SROM Accuracy for Changed Loading Conditions

Trunk nodes	Va Error (%)	Vb Error (%)	Vc Error (%)
701	-0.00177	-0.02971	0.088401
702	0.004375	-0.05148	0.153166
703	0.004777	-0.05171	0.153712
709	0.005019	-0.05178	0.154275
730	0.004941	-0.05176	0.154132
775	-0.02538	-0.03042	0.163088

3.3.1.7 Implementing SROM on the Arroyo-Hondo (A-H) Feeder

The A-H feeder model is a 6,534-bus radial distribution feeder with 11,641 single-phase nodes. It has a total of 4,350 Lines and 1,770 transformers. Of the total buses, there are 3,057 load buses that include 1-phase, 2-phase and 3-phase loads. The feeder structure is shown below in Figure 19.

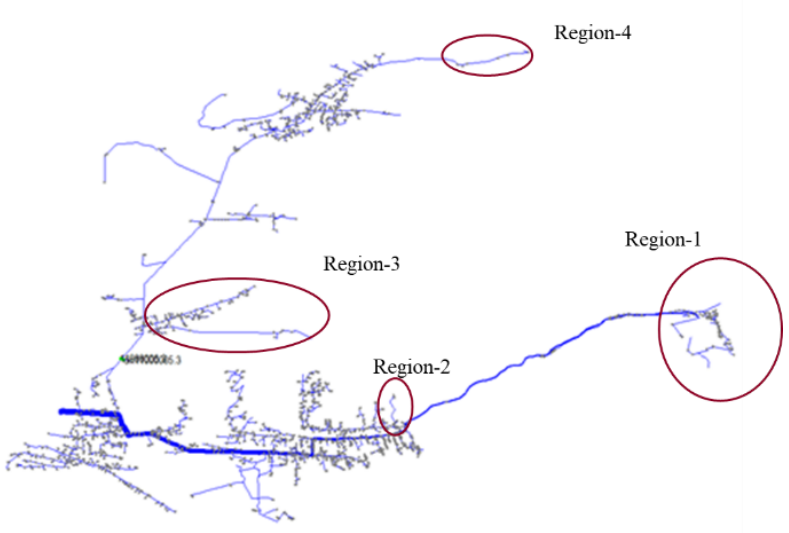


Figure 19: A-H Feeder model with the regions for aggregation indicated by red ovals

To prepare for the creation of the ROM, errors in the A-H feeder model were fixed and the fixes were informed by available AMI measurements. The OpenDSS model had some irregularities, and the details are given below:

1) The capacitances of most of the underground (UG) cables and the overhead (OH) lines in the feeder were very high. These were identified from the large negative reactive losses and overall negative reactive power demand at the substation. The power flow summaries from the OpenDSS models for the cases before the correction and after the correction are shown below:
2) Inaccurate voltage levels for the single-phase loads in OpenDSS. In OpenDSS, the single-phase loads should be specified with the nominal line-neutral voltage, but the model has line-line voltages. This caused an inaccurate power demand from the single-phase loads.
3) 50% of transformers had no downstream loads.
4) The voltage range for the loads were very wide with Vmax=2 pu and Vmin=0.7 pu.

Four regions of the A-H feeder, shown in Figure 19, that had good correlation to the latest GIS data are used to demonstrate the application of the SROM. These 4 trunk nodes were a mix of 1-phase, 2-phase and 3-phase nodes of varying scale as indicated in Table 12. The smallest region has 20 consumers, and the largest region has 243 downstream consumers.

The regions of the feeders that were to be aggregated into SROMs are shown in Figure 19. The identified regions were shown with red circles, and these are replaced by SROMs and after aggregation, these portions are replaced by the SROMs and were disconnected from the feeder. Once disconnected, OpenDSS detects these as hanging nodes.

Table 12: Statistics of the four regions

Region	Trunk Node for Each Region	Number of Phases	Number of Consumers
Region-1	OH2960022	3	243
Region-2	OH11000029	1	39
Region-3	OH430097	2	153
Region-4	OH2530068	1	20

3.3.1.8 S-ROM Implementation and Validation

The Camus Energy team helped to develop a network exploration tool based on Network-X and the GIS data of the A-H feeder model. The network exploration tool is utilized to determine all the downstream network components for any given node. This node is the one where the SROM will be connected to aggregate the downstream system. For the determination of the SROM parameters, the total losses and the total load downstream of a trunk node is needed.

3.3.2 Parameter Estimation

Consider a set of AMI measurements consisting of voltage *V*, real power *P*, and reactive power *Q* and the analysis described in the flowchart in Figure 20. When these power flows (P and Q) are used in the network power flow model to simulate the node voltages, these voltages may differ from the measured voltage in the AMI data. If these differences exceed a certain threshold, it may be due to inaccurate impedance parameters, which then need to be refined using distribution network parameter estimation (DNPE).

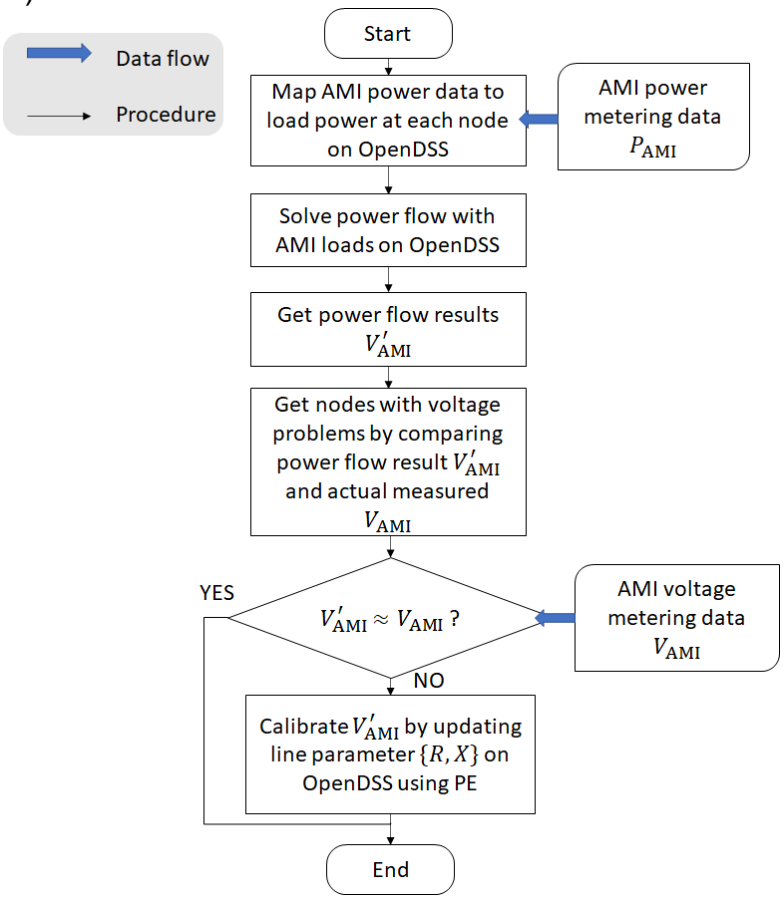


Figure 20: Map AMI metering data to power distribution system

The objective of distribution network parameter estimation (DNPE) is to use high-resolution AMI measurements of *P*, *Q,* and *V* to correct the *R*, X_L, and X_C parameters of a set of secondary distribution lines at the sites with high-resolution AMI data (see the red highlighted branch in Figure 21). V_u and V_d are the voltages at the upstream down-stream nodes, respectively, of these secondary lines. The ML approach used here estimates the optimal impedance parameters using three possible methods--linear regression algorithm, back-propagation neural network (BPNN) and long and short-term memory model (LSTM).

3.3.2.1 Training Data Generation

The training of ML-based DNPE models, especially the BPNN, requires a large amount of high-quality data. In the best data situation, these training data come from accurate AMI measurements under a wide range of operating conditions on networks with

accurately known impedance parameters. For the A-H distribution system and most distribution networks, such data are not available or are highly uncertain. Instead, we use power flow simulation to create these training data.

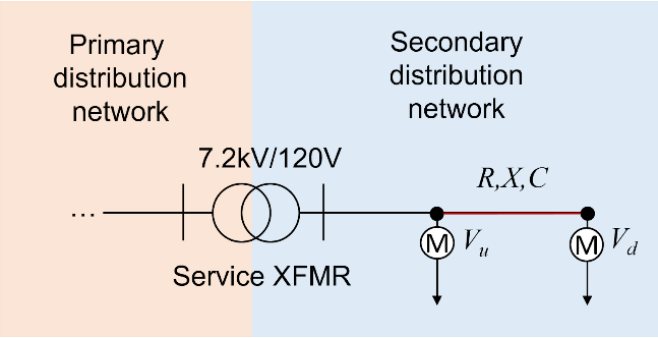


Figure 21: The low-voltage distribution lines addressed by DNPE algorithms

The initial distribution model that seeded the process in Figure 22 is the same A-H model. The impedance parameters *X* and *R* of the secondary distribution lines in this model are varied (via random sampling) and the power flow model is solved for the AMI data *V*, *I*, *P*, and *Q.* From these simulation samples, the line parameters are the labels, and the AMI data are the inputs for model training.

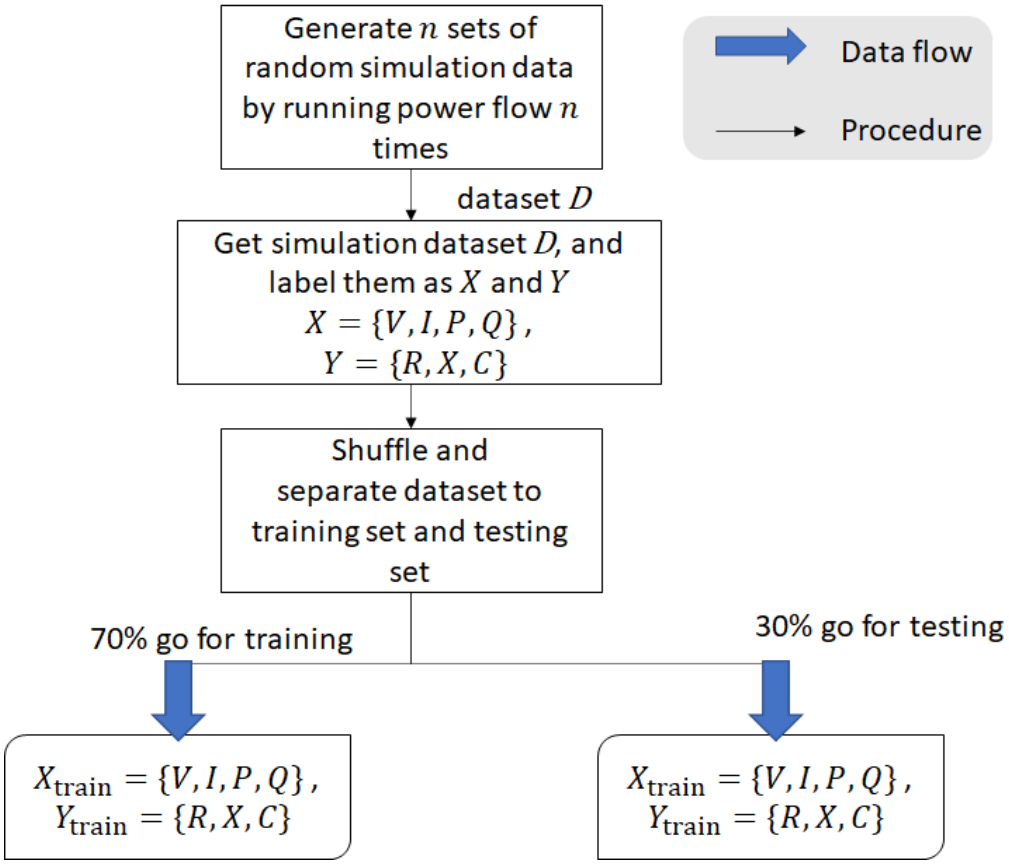


Figure 22: Sampling and power flow simulation process used to create training and test data for the DNPE algorithms

3.3.2.2 Offline Training

The offline training process is shown in Figure 23, and the specific process is as follows:

1. Step 1: Prepare the training data set from the previous process by splitting it into 70% for training ($X_{PE\text{-}train}$, $Y_{PE\text{-}train}$) and 30% for testing ($X_{PE\text{-}test}$, $Y_{PE\text{-}test}$)
2. Step 3: Construct the ML based DNPE model *f* and specify the model parameters.
3. Step 4: Use the training dataset to train the model *f* and create model *f**.
4. Step 5: Evaluate the model *f** using the withheld test data set ($X_{PE\text{-}test}$, $Y_{PE\text{-}test}$)
5. Step 6: Based on the evaluation, terminate the training or iterate the training until its accuracy within a preset threshold.

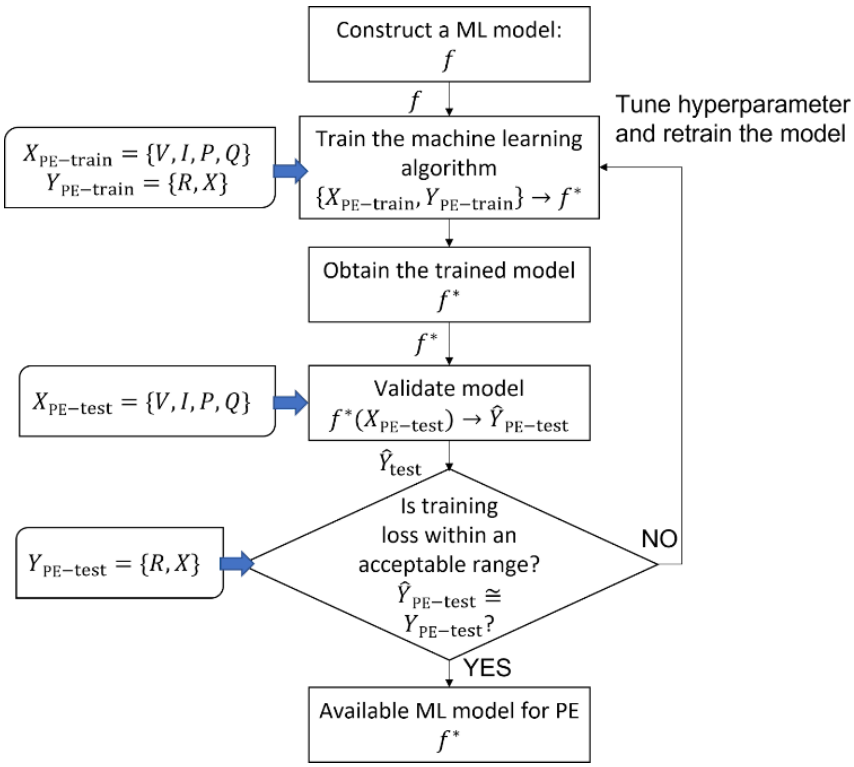


Figure 23: Offline training of ML-based DNPE model

3.3.2.3 Online Estimation of Pre-Trained DNPE Model

The pre-trained DNPE model is used for distribution line parameter estimation using the process shown in Figure 24:

1. Step 1: Solve power flow model using the measured AMI data for P.
2. Step 2: Traverse the entire distribution network to find the nodes where the difference between the simulated and measured voltages is larger than a prescribed threshold.
3. Step 3: Normalize the data at the nodes from Step 2 and serve the data as input to the trained DNPE model *f**.
4. Step 4: Use DNPE model f* to compute a set of new impedance parameters.
5. Step 5: Update the impedance parameters in the original power flow model and repeat as additional AMI data become available.

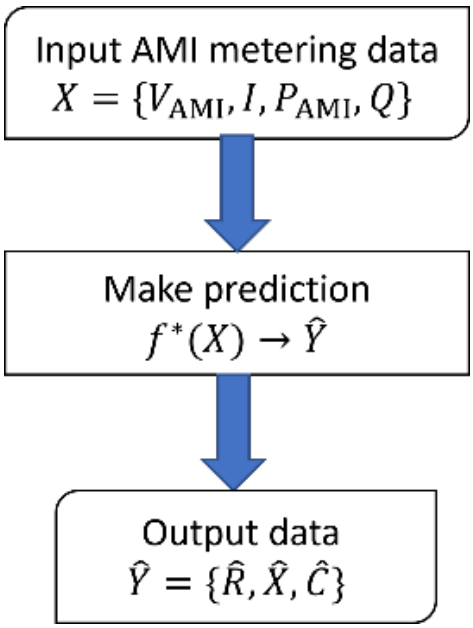


Figure 24: Estimate parameters using pre-trained model

3.3.2.4 Results

A DNPE model training and testing dataset containing 10,000 frames of samples was created. The range of the impedance parameters *R* and *X* and the range of the output parameters *U* and *I* are shown in Table 13.

Table 13: DNPE Model Training Data Generation

	Parameters	Range	Interval * Samples
Simulation Input	R (Ω)	1.345 – 6.345	0.05 * 100
	X (Ω)	0.5124 – 5.5124	0.05 * 100
Simulation Output	U (V)	110.015 – 130.97	-
	I (A)	0.15 – 12	-

Table 14 shows a subset of the results (for two overhead (OH) lines labeled “A” and “B”) from applying the DNPE approach to the A-H substation circuits. Prior to applying DNPE, the compute node voltages were 2-4 volts different (out of roughly 120 volts) from the measured AMI data. These differences are significant and would modify both the operational and planning decisions for these circuits. After applying DNPE and refining the *R* and *X* values in the secondary circuits, the simulated and measured voltage differences are within about 0.2 volts. Using our DNPE approach, it is possible to estimate correct line parameters at various locations with less computational burden than model-based methods.

Two metrics were used to evaluate the performance of the developed ML model – Mean square error (MSE) and mean average error (MAE). The metric MSE measures the variance of the residuals. The calculated MSE is 0.0328, and MAE is 0.179, which shows the estimated parameters from the proposed model are close to actual values.

Table 14: Voltage magnitudes seen in 2 overhead lines with and without parameter estimation

	Pre DNPE	Post DNPE	AMI
OH Line A	126.425	122.702	122.5
OH Line B	131.675	129.343	129.5

3.4 Task 4: Software Integration

Task 4 resulted in the integration of customer load and short term PV forecasts (subtask 4.2), and intraday PV forecasts (subtask 4.3) as described in Sections 3.4.2 and 3.4.3 of this document respectively. Efforts to fully integrate the power flow simulator (subtask 4.1) and the network parameter estimation (subtask 4.4) were unsuccessful; these efforts are described in Sections 3.4.1 and 3.4.4.

Figure 25 shows an architectural diagram for the forecasting components that were integrated into the larger Camus system. The forecasting system uses the ingestion pipeline as input data and produces the load and generation forecasts used by the physical models. The PV Process and AMI Process along with the HRRR Data Fetcher were implemented as part of the project. These composable Time Series Model components were integrated in the framework and deployed using the Camus Forecast System cloud architecture shown below. The entire Time Series Models library was released as part of the project. The cloud service architecture is representative of the broader Camus owned system. The emphasis on production system engineering you can see here is characteristic of our approach to these systems, focusing on speed and scalability.

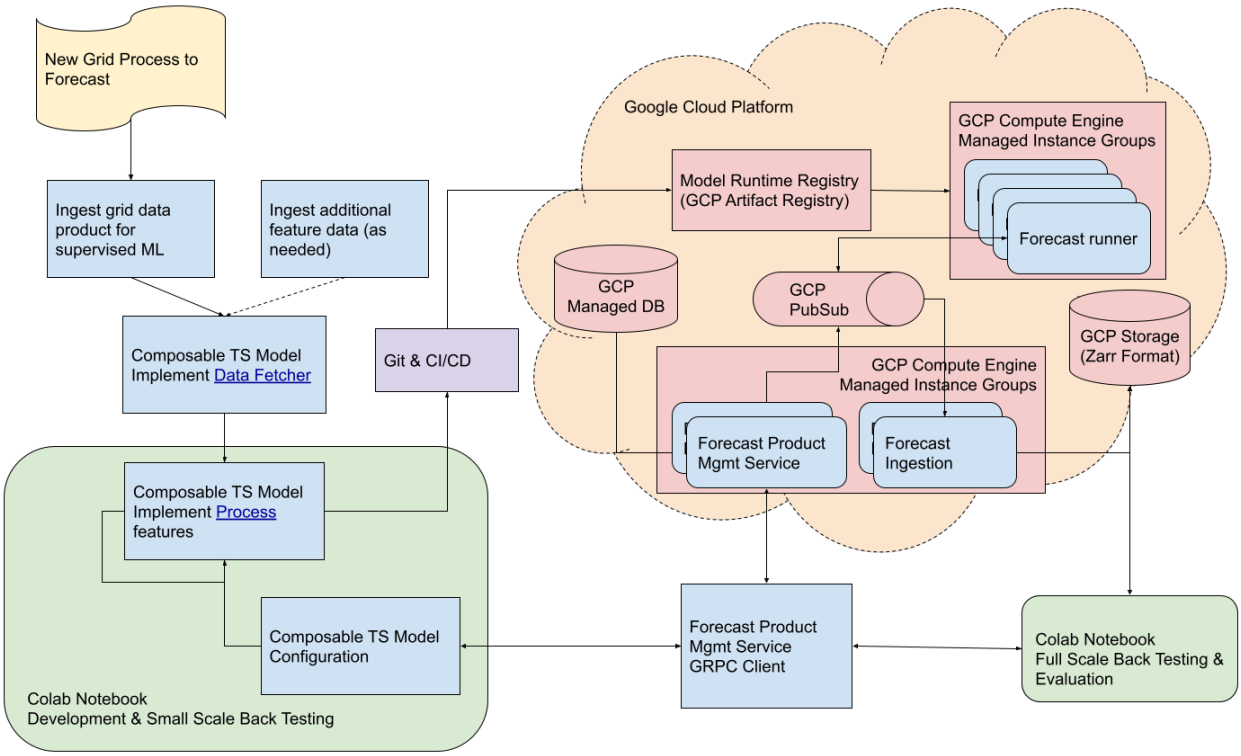


Figure 25: Architectural diagram for forecasting components in Camus Energy system

Additionally, a summary of algorithms used in Task 4 are described in Table 15 along with input data and assumptions. Camus's load forecast system uses supervised machine learning, a type of artificial intelligence that can learn complex behaviors from previous observations to make future predictions. Specifically, we are using XGBoost for supervised machine learning. This is an open-source library of "boosted tree"

models, which combine decision trees in ways that strengthen predictions while limiting overfitting. A decision tree on its own is generally not good at prediction, but a series of decision trees can be quite powerful, with each tree correcting the errors of the previous one. We have found that XGBoost strikes a good balance: It is fast and efficient, it is good at learning nonlinear relationships among the input features, and the results are explainable.

Table 15: Summary of algorithms used in Task 4

	Algorithm Used	Input Data & Assumptions
Gap Filling	Linear interpolation & imputation	AMI data; drop missing observations in training sets
Load Forecasting	XGBoost on autoregressive features & endogenous data; cohorting	AMI data, SCADA data, GIS, weather data, attributes (i.e. rate class, DER), seasonality
PV Forecasting	Physics based PV models	NOAA HRRR, PV configuration
Reduced Order Model	Machine Learning Based Network Parameter Estimation using AMI data	Feeder model, AMI
Ditto improvements		Feeder model, load snapshot

3.4.1 Power Flow Simulator Integration

Our goal was to integrate GridAPPS-D (or other power flow model) into our software pipelines to be able to run it at scale. We containerized the power flow simulator and integrated it into our systems using a python-based API. We performed validation tests on the IEEE 4-bus distribution test network. And we ran it in our systems as a quasi-static time series (QSTS) simulation that automatically advanced through consecutive timesteps, solving at each step and retaining its quasi-static state. Power flow inputs (load/generation using AMI usage and actual PV production) were applied at each note as it advanced through time.

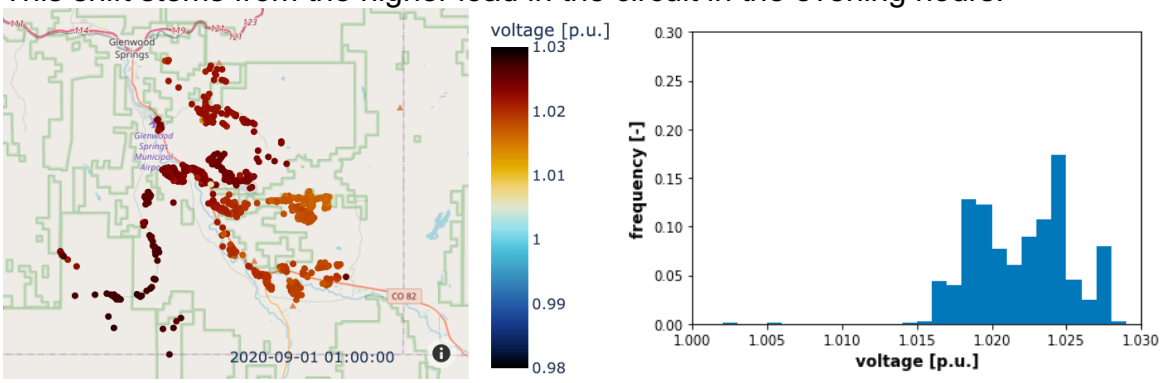
Despite these efforts, we struggled to run the simulator at scale. We found that converting the models into OpenDSS format was difficult and required modifications to NREL's DiTTo python package. Camus Energy submitted multiple contributions to the

NREL Ditto project to enable the GRIDLAB-D to OpenDSS workflow. These contributions are publicly available and documented on GitHub[6].

After exploration of the Kit Carson data set and power flow model, we determined it had significant missing data. We decided it would be more effective to perform this integration on another distribution cooperative's system. We chose a feeder because of reasonable completeness of AMI data at the service meter endpoints that had approximately 2700 meters and 1250 distribution transformers. As is, the OpenDSS power flow model we received had several issues. The model was incomplete and load nodes at the ends of secondary lines were missing. The model was also plagued with inconsistencies with several key fields such as phasing, power ratings, and voltage bases of distribution transformers often not in agreement with GIS data. Anecdotally, this is not surprising: power flow models are not updated as frequently as the distribution grid inventory and are more often than not out of sync with GIS data. To resolve these discrepancies, we modified OpenDSS files based on GIS inputs. For instance, we created *Loads.dss* representing all the load nodes in the system, aggregated at the distribution transformer level. This process was tedious and painstaking as OpenDSS (or any power system tool) relies on consistency between various elements in the power system for convergence.

We developed a power flow simulation tool which interfaces with OpenDSS via *opendssdirect* an official python extension of OpenDSS. Given OpenDSS files and AMI data, this tool solves snapshots sequentially and records metrics for post-processing.

The following Figure 26 shows voltage distributions across the feeder at two distinct time instances. At 01:00 hrs local time, the majority of bus voltages (per unit) are in the range of 1.015-1.030, while this distribution shifts to 1.005-1.015 at 19:00 hrs local time. This shift stems from the higher load in the circuit in the evening hours.



[6] https://github.com/nlaws-camus/ditto

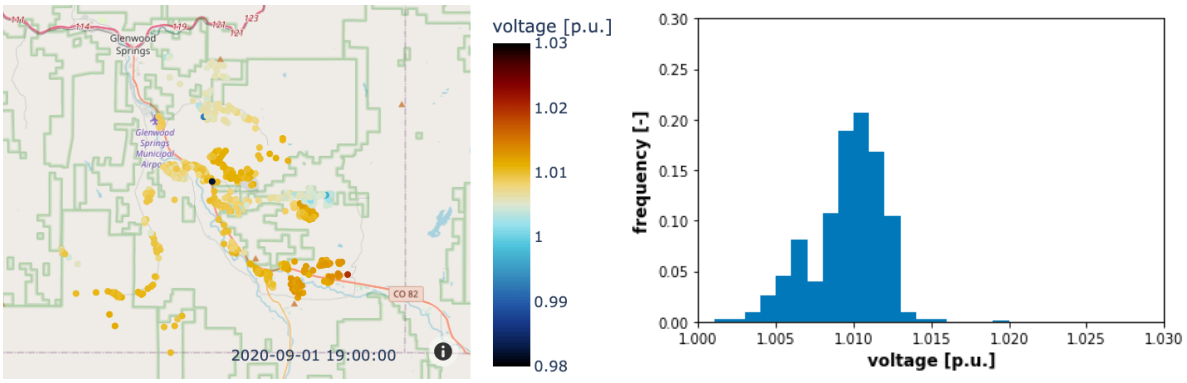


Figure 26: Voltage distributions at two distinct time instances

It is also interesting to see the diurnal variation in bus voltages in different parts of the feeder. In the following Figure 27, bus 6433t7 is farther towards the end of the lines and shows relatively exaggerated voltage excursions compared to the more centrally situated bus 6550t9.

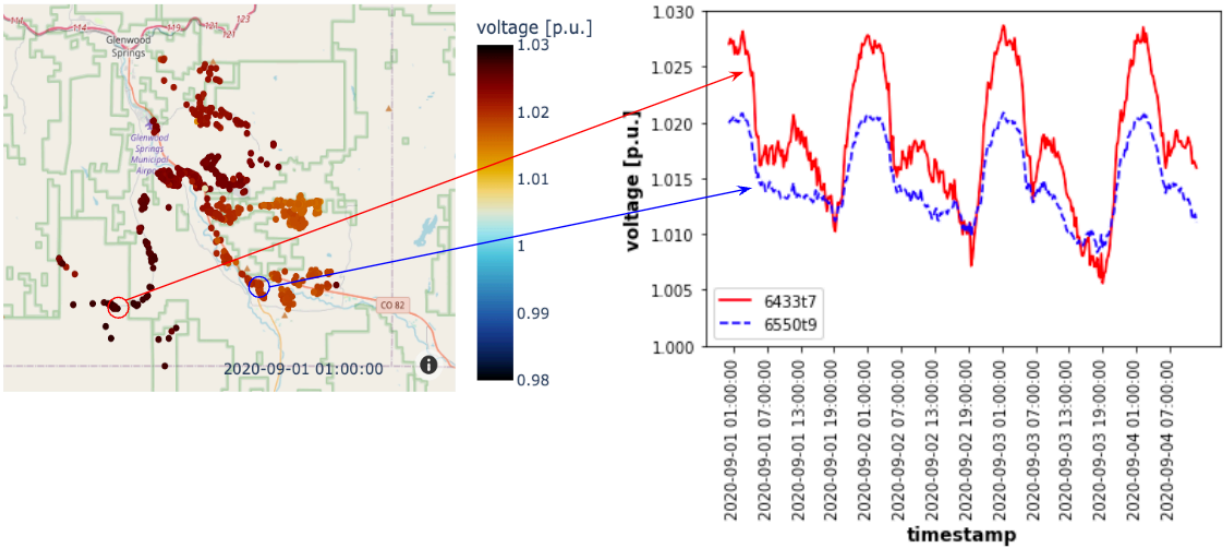


Figure 27: Diurnal variation in bus voltages in different parts of the feeder

3.4.2 Customer Load and Short-Term PV Integration

We integrated the highest performing ML methods from Task 2.1, including parallelization and performance tuning, to meet real-time reporting requirements.

We select the inputs (or “features”) drawing on our collective knowledge and experience. We tune these inputs for each forecast system, driven by the unique characteristics and behaviors of each distribution grid. The forecast model learns the relationship between the inputs and the forecasted net and gross load values. The result is a “trained” model which is later used to make predictions based on new inputs

and the learned historical behaviors. Features include harmonics, day of week, lags, temperature, humidity, and downward shortwave radiation flux.

Cohort-based modeling enables us to leverage all the available data to inform our forecast at each individual load point without incurring massive compute costs. We use metadata to divide load points into cohorts with similar load behavior. This enables the forecast system to learn the load behaviors from data across many substations without the complexity and cost of distinct models. We then use each cohort model to make individual predictions for every load point, allowing the forecast to adapt to changes.

Customer load and short-term PV forecasts solve several challenges for distribution utilities. First, it allows for gap-filling of data in the case data is missing, as seen on May 29 in Figure 28.

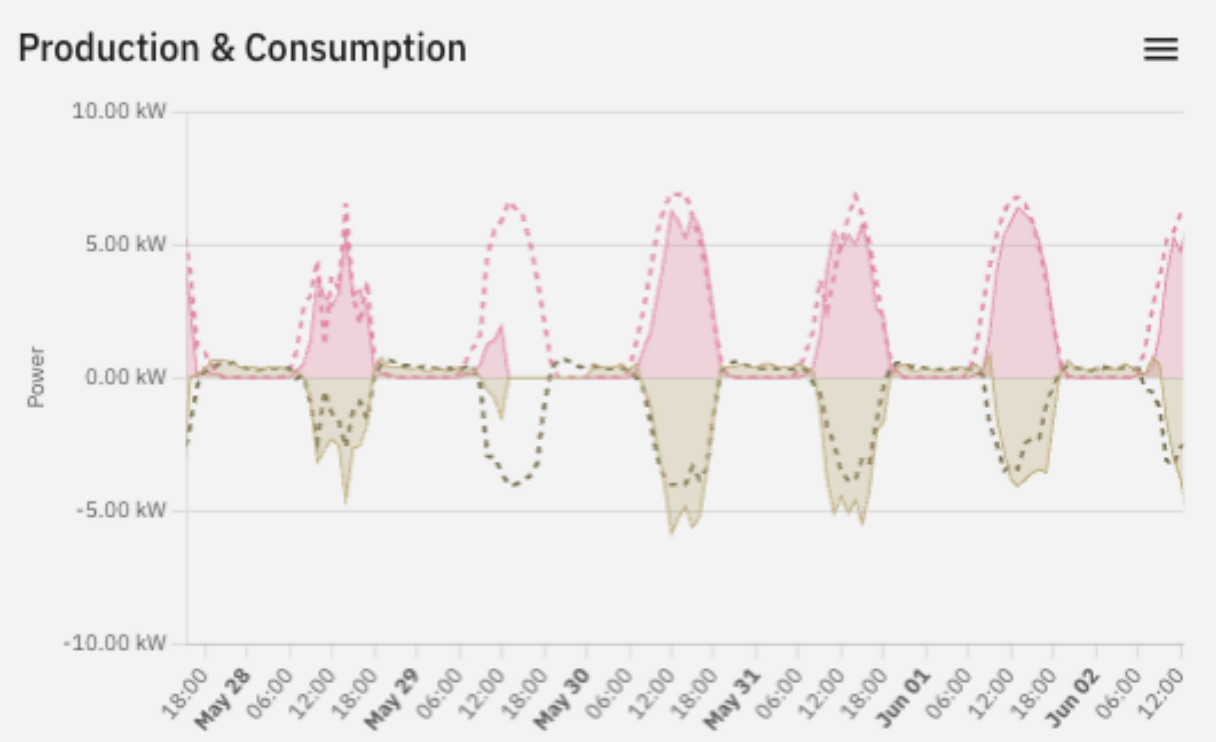


Figure 28: Gap filled generation and production data

Second, it allows for the estimation of solar PV production for systems that do not have a production meter, as seen in Figure 29:

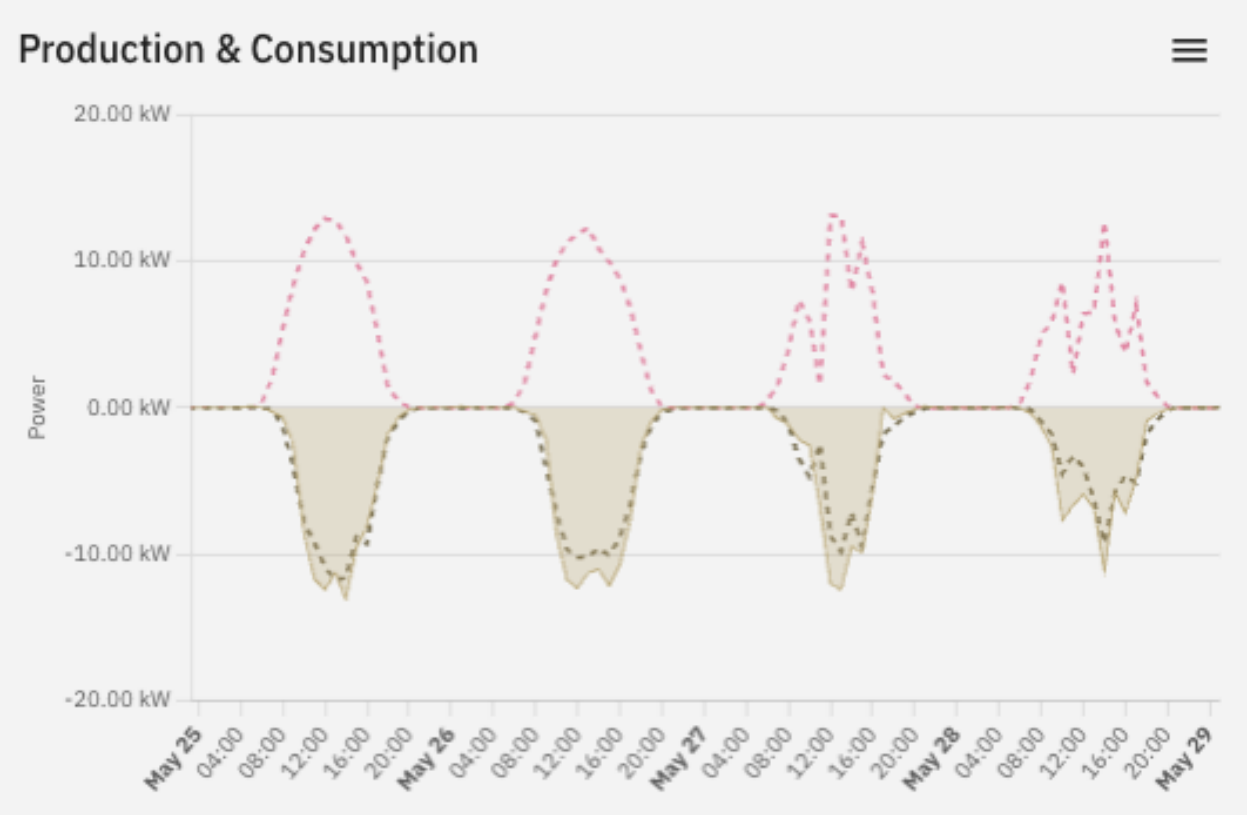


Figure 29: Estimated generation at non-metered PV system

Third, it closes the collection gap between energy consumption and production, and when those readings are available to various systems. For example, in Figure 30, data from June 14 may not be available until June 16, but the nowcasts can be used to close this gap, making the data available in real time.

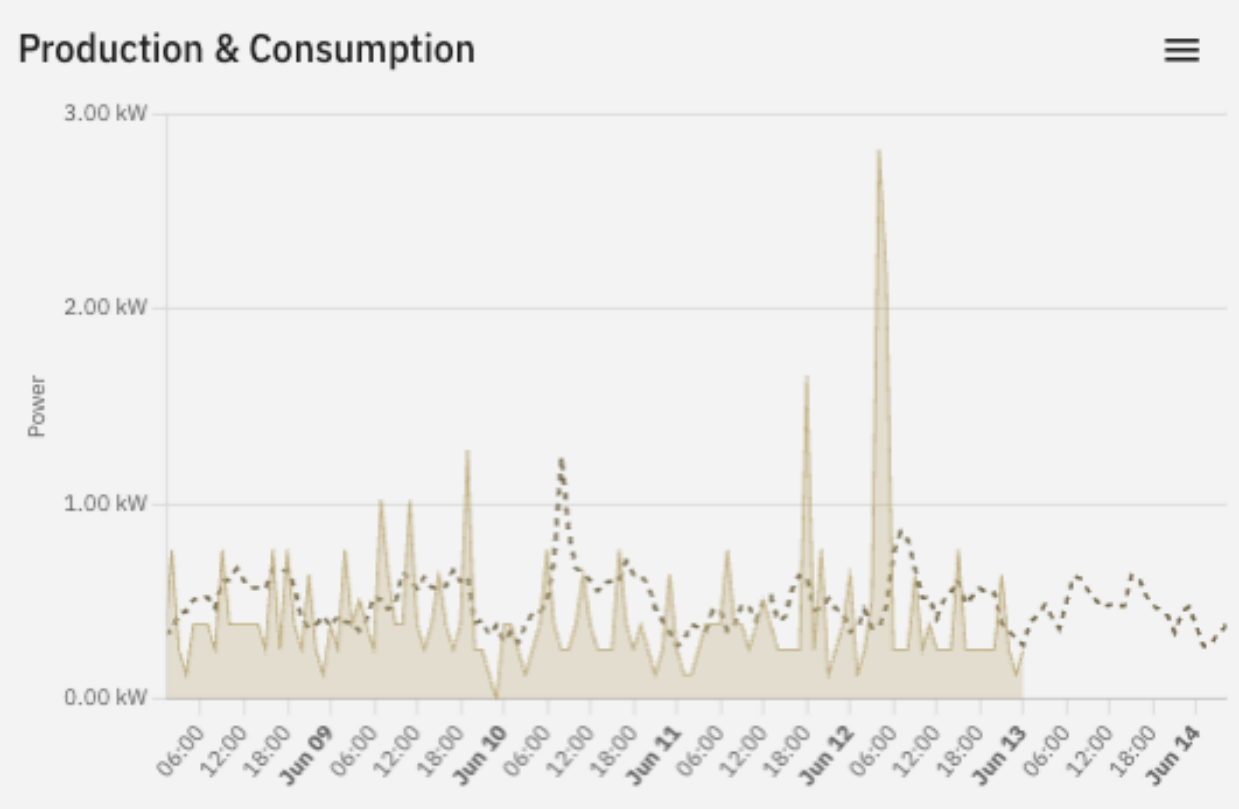


Figure 30: Closing data collection gap

3.4.3 Intraday PV forecast integration

We completed the infrastructure work to support integration of load and forecasting models into the Camus testing environment, including the assessment of computational requirements, development of parallel computing approaches, and validation and demonstration of gap filling on historical AMI data.

The Camus PV forecasts leverage information about the physical system (such as the AC and DC capacity of the system, tilt, and orientation) along with weather data forecasts from the National Oceanic and Atmospheric Administration (NOAA). The NOAA weather forecasts, available directly in the cloud through the NOAA Open Data Dissemination program, are a key input in our forecast system. These high resolution weather forecasts update every hour for an 18-hour forecast horizon, and every 6 hours for a 48-hour forecast horizon. NOAA developed its High Resolution Rapid Refresh (HRRR) model to help predict renewable energy generation, and it assimilates data from satellites, radar, ground stations, and weather balloons.

These intra-day PV forecasts along with AMI forecasts as shown in Figure 31, allow utilities to do better short term planning.

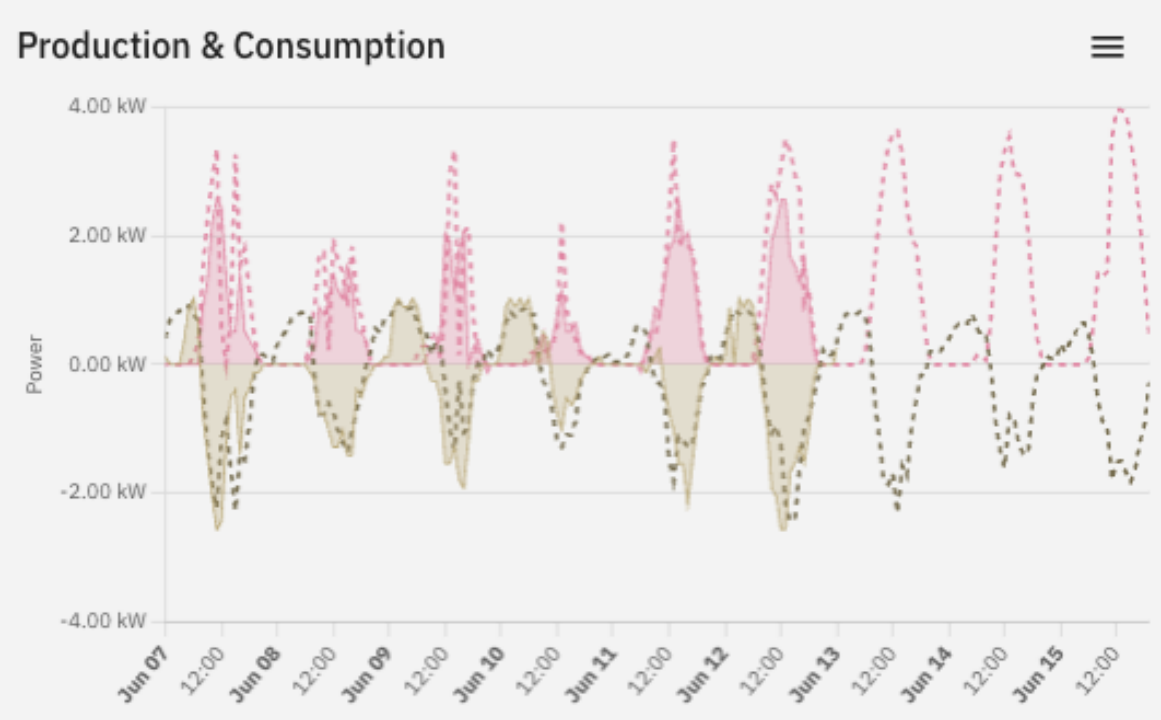


Figure 31: Intra-day PV and AMI forecasts

3.4.4 Network Parameter Estimation

We were not able to implement a repeatable methodology for network parameter estimation because of the difficulties associated with calibrating incomplete or dated network models. We were not able to integrate a systematized methodology into the software platform. This, in turn, made completing all of Task 5 subtasks equally challenging.

3.4.5 Open Source Access to Code

In addition to integrating short term load and PV forecasts into the Camus platform, as part of this award, Camus made the code publicly available per the End of Project deliverable, via an Apache 2.0 license on a GitHub repository[7], with data available on a Google Cloud Storage bucket. Code will run most effectively on a Linux box but is not coupled to any one operating system. The code includes a readme file[8] with an overview of the code. A summary of the readme file is provided below.

This package provides the tools to construct machine learning models that fill gaps or forecast in the verification datasets. This includes loading the data and applying the XGBoost estimator. The composable model framework provides the configurable model inputs required for the Neighbor Informed Estimates and Community Analytics. The PV System model is also implemented using the composable model framework.

[7] https://github.com/SETO2243/forecasting

[8] https://github.com/SETO2243/forecasting/blob/main/README.md

The API for fit, predict, and metrics is reduced to specifying a start and end times for a given location. The model must construct feature data using column transforms. Having done so, forecasting as a service becomes trivial.

This library is designed for use by technical engineers and data scientists. It takes advantage of the Python data science ecosystem and therefore requires installation of many third party open source libraries. It has been developed and tested in a Linux operating system.

Models can be composed of mixins for various estimators and forecast processes. These composable pieces can be put together in different ways to solve many problems. The RegularTimeSeriesModel is the core that problem specific parts are added to when forecasting or gap filling a particular time series. The estimator is the next essential building block. The estimator can be either a Classifier (a discrete estimator) or a Regressor (a continuous estimator). There are many different numerical techniques for supervised learning estimators. The process is the last essential component. It defines the time series being forecast and the available feature data that might have predictive value. Having composed a Model class from these three parts, it is then up to the user to create an instance of the class with configuration arguments that tune the model features for the specific meter load or PV forecast.

The PV Model uses the same composable framework to define models using the HRRR weather (see below) as an input to the NREL PySam PV generation algorithm. For the project we used the PySam generation forecast directly using the configuration shown below with the IdentityRegressor. Building additional input features for sites with direct telemetry would allow using machine learning models like XGBoost too.

3.5 Task 5: Integrated Software Performance Verification

The goal of task 5 was to test and validate the performance of the software modules that were integrated into the Camus platform. The SOPO called for the validation against three different states of the network model: Subtask 5.1 (Approach #1 performance verification), where a network model is available and accurate, Subtask 5.2 (Approach #2 performance verification), where the network model is available but the parameters are inaccurate, and Subtask 5.3 (Approach #3 performance verification,) where the network model is not available.

Because of the challenges with quality in the customer sourced power flow model and extra effort spent on correcting these models along with contributing to enhancements to the Ditto source code (as described in detail in Section 3.4.1), we were never able to fully integrate the power flow model into the Camus infrastructure. As a result, we were not able to evaluate the performance of the ML/DA generated data against either the available and accurate model in approach #1, nor the inaccurate model in approach #2. However, we were able to assess the performance of the ML/DA generated data against actual meter endpoint data collected from the utility as described in approach #3.

3.5.1 Approach #1 performance verification

The project team was unable to evaluate the ML/DA approaches against an available and accurate model because such a model was not made available.

3.5.2 Approach #2 performance verification

The project team was unable to evaluate the ML/DA approaches against a model with inaccurate parameters due to excessive inaccuracies in the model leading to the inability to converge, as well as challenges in model conversion to open source software.

3.5.3 Approach #3 performance verification

Analysis of the neighborhood informed analytics compared the gap filling outputs of the XGBoost algorithm with the true data. The verification process included data from a full month of operations. Data was collected at 15-minute intervals for the entire month and produced 2,976 instances. To represent missing data, portions of the set were removed and in total equaled about 500 missing data points. Figure 32 shows the total data set for one of the 20 meters used in the verification experiment. The blue sections in Figure 32 depict the available data, while the red indicates where data was missing. The available and unavailable data depicted in Figure 32 represented a situation where a meter produced unreliable data.

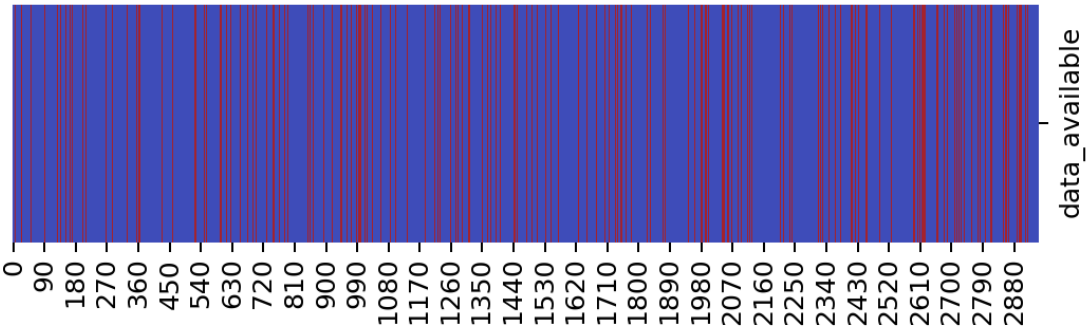


Figure 32: This image depicts the data set for the verification process where the red lines indicated randomly removed gaps in the data and the blue depicts the available data.

The neighborhood informed analysis approach proved to provide reliable estimates of missing data. For example, applying the estimation approach for the meter highlighted in Section 3.2.1.3.2 produced results that were close to actual. Figure 33 shows a snapshot of the time series data for this meter. The collected data is in blue, which includes gaps that are filled in by the true (or actual) value using the gray dashed lines. The gap filling results are depicted with the red circles. These results did not fall exactly on the actual line each time but provided a reasonable approximation.

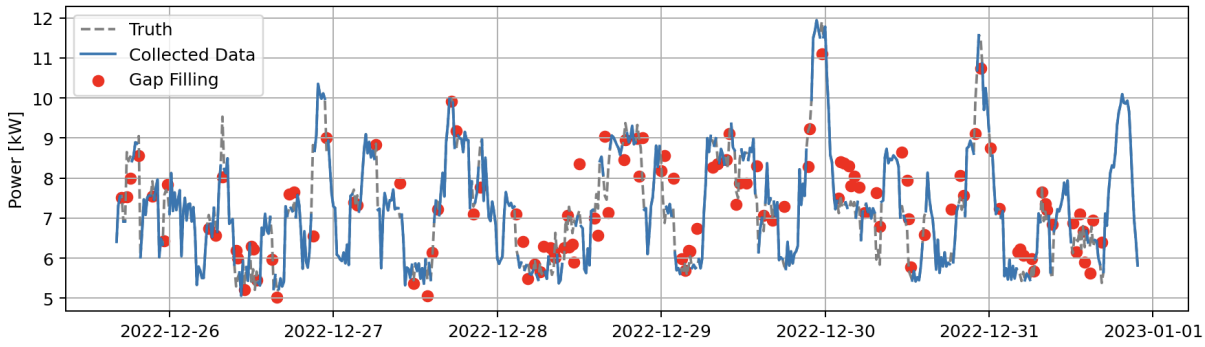


Figure 33: Sample time series plot of the gap filling for a single meter inside its group

To quantify the gap filling estimate accuracy for this meter, Figure 34 plots the estimated versus actual. The results show that the model had a reasonable, but not great, level of fit. The r-squared value was computed to be 0.65. A visual inspection of the comparison indicated that the model tended to over predict at low power and under predict when the actual power was high.

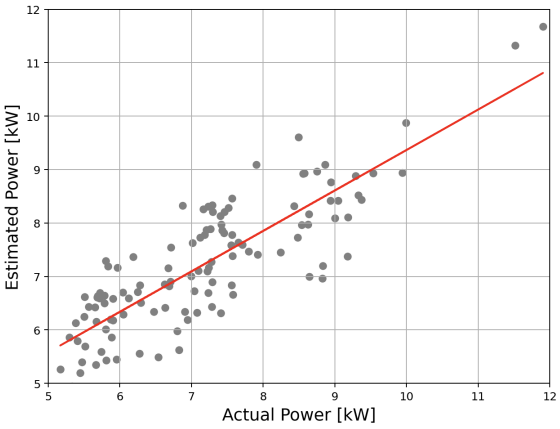


Figure 34: Meter 12's actual versus estimated power comparison for the single meter. The least-squares linear regression produced an r-squared of 0.65

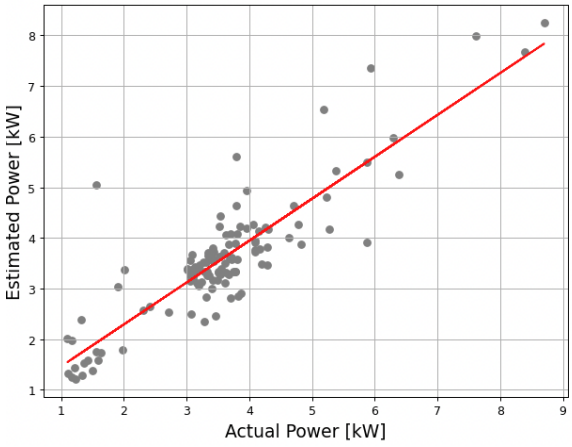


Figure 35: Meter 17's actual versus estimated power comparison for the single meter. The least-squares linear regression produced an r-squared of 0.74

The estimation results varied for the 20 different meters used in this experiment. For example, the analysis of data from another meter located in a different section of the grid produced a slightly higher r-squared value: 0.74. The estimated versus actual results, and the least squares linear fit line are shown in Figure 35.

The overall results for all 20 meters are shown in Figure 36, which indicates that on average the approach was able to produce very good results. The r-squared value for the least squares fit of all the metes was 0.92. The least squares line, depicted with the red line in Figure 36, indicated that the estimate had very little bias throughout the entire range of actual power values. At low power, which was between 0 kW and 6 kW, the estimate tended to provide an accurate representation of the system. Similarly, at high

power draws above 6 kW, the estimated power tended to be correct. However, above 10 kW the estimate tended to underestimate the actual value slightly.

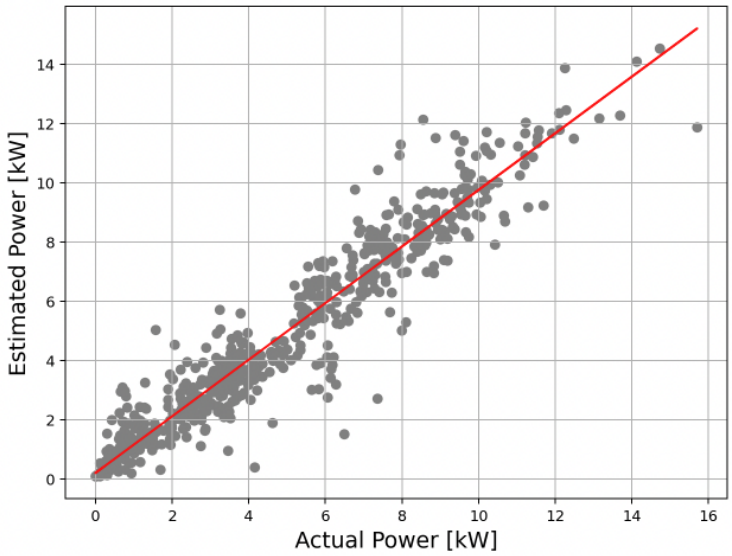


Figure 36: The combined actual versus estimated power comparison for all the meters. The least-squares linear regression produced an r-squared of 0.92

4 Significant Accomplishments and Conclusion

The project found that ML can empower utilities by improving grid awareness in the following areas:

1) Fill in missing data gaps for more complete visibility of system states.
2) Forecasting PV generation for more informed gap filling.
3) Estimate model parameters for efficient physics-based modeling.

The report describes the implementation of two gap filling techniques. Each technique used the XGBoost ML algorithm but performed the training and testing tasks differently. In some cases, the ML algorithms used here did not outperform standard approaches but used least compute power and could be deployed with little effort. Validation of the neighborhood ML gap filling approach resulted in high r-squared values. This indicated that the ML approach had a strong “goodness-of-fit”.

Weather forecasts and AMI data effectively informed a PV estimation model useful for predicting generator outputs.

Extensive work was done to generate ROM of an IEEE and an actual system. The error results of the model were low and indicated that scaling the method will be useful for efficiently modeling system states during current or future conditions. And parameter estimation methods using ML can effectively identify the appropriate parameters for modeling an electrical system accurately.

5 Path Forward

Future work that builds on this report will include an expansion of the gap filling analytics. One area for immediate improvement is the implementation of the cohort training and testing approach. Other research and testing are necessary to test the hypothesis that it will produce more accurate results when training includes proper filter of input data to exclude gaps or erroneous data.

Also, future work can expand on the ROM work to further develop, test, and validate the approach.

6 Products

C. Qin, B. Vyakaranam, P. Etingov, M. Venetos and S. Backhaus, "Machine Learning Based Network Parameter Estimation Using AMI Data," *2022 IEEE Power & Energy Society General Meeting (PESGM)*, Denver, CO, USA, 2022, pp. 1-5, doi: 10.1109/PESGM48719.2022.9917034[9].

[9] https://www.pnnl.gov/publications/machine-learning-based-network-parameter-estimation-using-ami-data

7 Project Team and Roles

Principle Investigator: Cody Smith

Project Management: Sydney lineman, Emma Elggvist, Dylan Cutler

Engineering Analysis: Nick Laws, Akhilesh Bakshi, Michael Hutson, David Stuebe, Milton Venetos, C. Birk Jones, Alok Kumar Bharati, Bharat Vyakaranam, & Pavel Etingov

Business Support: Michael Ryan, Raj Raheja, and Scott Backhaus

8 References

[1] D. Papale, "Data Gap Filling," in *Eddy Covariance*, Springer, 2011, pp. 159-172.

[2] P. Arriagada, B. Karelovic and O. Link, "Automatic gap-filling of daily streamflow time series in data-scarce regions using a machine learning algorithm," *Journal of Hydrology,* vol. 598, 2021.

[3] Z. Charles, Z. Garrett, Z. Huo, S. Shmulyian and V. Smith, "On Large-Cohort Training for Federated Learning," in *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, 2021.

[4] A. R. Matavalam, A. Singhal and V. Ajjarapu, "Monitoring Long Term Voltage Instability Due to Distribution and Transmission Interaction Using Unbalanced PMU and PMU Measurements," *IEEE Transactions on Smart Grid,* vol. 11, no. 1, pp. 873-883, 2020.

[5] M. Netto, Y. Susuki, V. Krishnan and Y. Zhang, "On Analytical Construction of Observable Functions in Extended Dynamic Mode Decomposition for Nonlinear Estimation and Prediction," *IEEE Control Systems Letters,* vol. 5, no. 6, pp. 1868-1873, 2021.

[6] Q. Huang and V. Vittal, "Integrated Transmission and Distribution System Power Flow and Dynamic Simulation Using Mixed Three- Sequence/Three-Phase Modeling," *IEEE Transactions on Power Systems,* vol. 32, no. 5, pp. 3704-3714, 2017.

[7] A. Bharati and V. Ajjarapu, "Investigation of Relevant Distribution System Representation With DG for Voltage Stability Margin Assessment," *IEEE Transactions on Power Systems,* vol. 35, no. 3, pp. 2072-2081, 2020.

[8] V. Ajjarapu, V. Umesh, M. Govindarasu, V. Krishnan, A. R. R. Matavalam, M. Netto, A. K. Bharati, P. Sharma and B. Huang, "Sensor enabled data-driven predictive analytics for modeling and control with high penetration of DERs in distribution systems," USDOE Office of Electricity (OE), 2021.

[9] J. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics,* vol. 29, no. 5, pp. 1189-1232, 2001.

[10] C. E. Commission, "Solar Equipment List," 1 1 2023. [Online]. Available: https://www.energy.ca.gov/programs-and-topics/programs/solar-equipment-lists.