PNNL-38890

VRN3P: Variational Recurrent Neural Network Based Net-Load Prediction under High Solar Penetration

Final Technical Report

January 2026

Soumya Kundu, PNNL
Allison M. Campbell, PNNL
Kaustav Bhattacharjee, PNNL
Orestis Vasios, PNNL
Andrew P. Reiman, PNNL
Indrasis Chakraborty, LLNL

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, **makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights**. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from
the Office of Scientific and Technical
Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fox: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov

Final Technical Report (FTR)

Cover Page

a. Ferderal Agency	*Department of Energy*	
b. Award Number	*37774*	
c. Project Title	*VRN3P: Variational Recurrent Neural Network Based Net-Load Prediction under High Solar Penetration*	
d. Recipient Organization	*Pacific Northwest National Laboratory (PNNL)*	
e. Project Period	Start: *02/01/2021*	End: *09/30/2023*
f. Principal Investigator (PI)	*Soumya Kundu, PhD* *Senior Research Engineer* *soumya.kundu@pnnl.gov* *509-375-2431*	
g. Business Contact (BC)	*Sherry Kowalski* *Senior Project Coordinator* *sherry.kowalski@pnnl.gov* *509-372-4045*	
h. Certifying Official	*Same as the PI*	

Signature of Certifying Official   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.

Acknowledegment

This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the DE-FOA-0002243 Award Number 37774.

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Protected Rights Notice

This protected data was produced under agreement no. 37774 with the U.S. Department of Energy and may not be published, disseminated, or disclosed to others outside the Government until five (5) years from the date the data were produced, unless express written authorization is obtained from the recipient. Upon expiration of the period of protection set forth in this Notice, the Government shall have unlimited rights in this data. This Notice shall be marked on any reproduction of this data, in whole or in part.

Executive Summary

The goal of this project, led by Pacific Northwest National Laboratory (PNNL), in collaboration with Lawrence Livermore National Laboratory (LLNL) and Portland General Electric (PGE), was to develop and validate a deep variational recurrent neural network-based net-load prediction (VRN3P) framework for probabilistic time-series forecasting of day-ahead net-load under high solar penetration scenarios. Some of the key project objectives were to: i) systematically capture uncertainty and variability; ii) engineer deployability; and iii) enhance interpretability in the deep forecasting models. Comprehensive validation of VRN3P was performed on test-case scenarios developed in GridLAB-D using anonymized real-world solar, net-load, and end-use specific data, representative of varying high-solar scenarios (20-50% in annual energy). Moreover, PNNL-developed GRAF-Plan tool was used to construct a work-flow to estimate the relative reduction in balancing reserves procurement due to improvement in day-ahead net-load forecasts generated by VRN3P. Final project outcome includes an open-source validated probabilistic net-load forecasting tool VRN3P, made publicly available via an interactive user-interface.

The project team reports successful design of a novel probabilistic net-load forecasting architecture, comprising of a variational autoencoder and a recurrent neural network, which demonstrates 30% improvement in forecast performance, 60% improvement in training time, and consumes 44% less memory, when compared with conventional baseline models. The team tested the VRN3P model performance on GridLAB-D test-cases representing varying BTM solar penetration levels of 20%, 30%, and 50%, with integrated time-series net-load profiles provided by the utility partner (PGE). The VRN3P model demonstrate $<$2% hourly MAPE (averaged over the year) for day-ahead net-load forecast on the test scenario with 20% BTM solar. Transfer learning extension of the VRN3P model has demonstrated $8.33\times$ speed-up in training, while still achieving acceptable forecast performance of 2.24% hourly MAPE on the 30% BTM solar penetration test-scenario. A preliminary version of the VRN3P GridAPPS-D™has been developed, along with a web-based interactive user-interface (named ‘Forte’) which has made available on GitHub for public use.

Enhanced net-load forecast by the proposed VRN3P framework is expected to deliver *significant economic benefits*, such as reduction in non-spinning reserves requirement (by reducing net-load forecast errors), savings for ratepayers with improvement in the accuracy of net-load forecast; in addition to playing a critical role in meeting the *SunShot 2030 goal of 50% reduction in levelized cost* of (solar) energy.

Contents

1 Background

Reliable and accurate means of forecasting power generation and net-load are essential for maintaining an uninterrupted energy supply in the power grid. Forecasting becomes especially valuable in the context of utilization of renewable power, which is inherently dependent on atmospheric conditions and therefore uncertain. The uncertainty associated with renewable power sources such as wind and solar also increases the energy cost by requiring ancillary frameworks for load balancing during periods of unexpectedly high and low production [27]. Existing practices use an *additive* approach to *net-load* forecasting by subtracting the solar forecast values from the (independently obtained) load forecast values. In contrast, an *integrated* approach of directly synthesizing the net-load forecast via systematic combination of both solar and end-use variability, as proposed in this effort, is better suited for capturing the latent correlations in solar and end-use load patterns and thereby significantly enhancing the forecast accuracy [20, 39]. Unveiling the complex inter-dependency between solar and end-use load patterns driven by several latent factors (from meteorological to behavioral) [32] necessitates the development of a deep learning-enabled *variational inference* technique, similar to the one introduced in this project. In addition to the latent correlations, another key driver for net-load forecast uncertainty is its temporal variability, manifested in the form of *ramps*, i.e. rapid changes in the net-load. While some of these ramp events occur during the shoulder hours (beginning and end of the solar day), resulting in the *duck curves* [20], others are often triggered by geographically localized factors such as cloud cover. State-of-the-art methods that focus on solar-only ramps prediction [15] cannot accurately capture the net-load ramps which often display distinct temporal characteristics different than the solar-only ramps [8]. A key challenge in forecasting net-load ramps lies in learning the functional mapping between the temporal variations in solar and end-use load (e.g. heating/cooling) and the contributing micro-climatic factors (e.g. cloud cover, temperature), which this project set out to resolve via the implementation of a powerful deep recurrent neural network architecture.

Deep learning techniques, in particular recurrent neural networks (RNNs) based models, such as long-short-term-memory (LSTM) networks, have been used for solar and power forecasting with encouraging performance [16, 1, 44, 35, 43]. For instance, the authors of [44] used LSTM with attention mechanism for short-term (up to 60 min) solar power forecasts and achieved $\sim$20% MAPE. However, the model in [44] is deterministic and the work did not report the spread of errors to quantify the reliability and generalizability of the model. In a recent work [40], the authors proposed a convolutional layer to extract important features from the input window prior to an LSTM cell, thereby improving the training efficiency, and applied the concept to solar irradiation prediction. However, the predictions were deterministic and the uncertainty in the estimates was not studied. In [11], the authors compared five different benchmark probabilistic methods for an hourly solar forecasting task. But the work did not consider the problem of net-load forecasting in presence of behind-the-meter solar, which are driven by variability in end-use load behavior.

To our best knowledge, the approach adopted in this project in unique in its attempt at integrated probabilistic net-load forecasting with deep learning-enabled systematic treatment of the spatio-temporal impact of several latent contributing factors (meteorological, behavioral, or otherwise) on the variability and uncertainty of the net-load.

2 Project Objectives

In the following, we briefly describe the project goals, innovation, impact, tasks, and milestones.

2.1 Project Goals

The project team of two DOE national laboratories (PNNL and LLNL) and an electric utility (PGE) set out to develop and validate a deep variational recurrent neural network-based net-load prediction (VRN3P) framework for probabilistic time-series forecasting of day-ahead net-load under high solar penetration scenarios. Key project objectives are: 1) systematic capturing of uncertainty and variability; 2) delivering theoretical assessment of forecast performance; 3) engineering deployability; and 4) enhancing interpretability. Anonymized real-world solar, net-load, and end-use specific data, shared by utility partner (PGE), were used to synthesize representative grid models with high BTM solar penetration in GridLAB-D for comprehensive validation of VRN3P. FOA-specified target accuracy of 2% hourly MAPE will be demonstrated in different scenarios representative of varying high-solar scenarios (20-50% in annual energy) and seasonal and daily end-use variations. PNNL GRAF-Plan tool [13] were to be leveraged to estimate the impact of VRN3P on reduced reserves requirement. The final project outcome(s) would include an open-source validated probabilistic net-load forecasting tool VRN3P, with target forecast accuracy of 2% MAPE.

2.2 Innovation and Impact

Uniqueness and innovation: Existing forecasting practices typically involve independently generating solar-only and load-only forecasts, from which the net-load forecasts are synthesized in an additive manner. Such methods, unfortunately, are incapable of capturing the complex temporal inter-dependency and correlations in various latent (meteorological to behavioral) factors that drive the variability and uncertainty in net-load at high-solar penetration. To this effect, the proposed VRN3P is unique and novel because it is, to our best knowledge, a first attempt at developing a deep learning-based probabilistic net-load forecasting which systematically captures the spatio-temporal impact of high-dimensional factors (meteorological, behavioral or otherwise) on the variability and uncertainty of the net-load. The key innovations of the proposed framework are in: i) extension of the state-of-the-art point forecast methods to probabilistic forecasts with theoretically justified confidence intervals; ii) demonstrated performance improvement over the state-of-the-art by achieving 2% MAPE at over 20% solar penetration (in annual energy); iii) synthesis of efficient algorithms that bring together the power of deep learning methods and the practicality of limited computational resources available in field implementations.

Impact: Improvement in net-load forecasts can have significant economic benefit, in addition to several environmental benefits. An ERCOT analysis estimated around 5-40% increase in monthly average non-spinning reserves requirement due to net-load forecast errors [34], while a California Energy Commission (CEC) report estimates savings worth over $2M/year for CAISO and California ratepayers with 5% improvement in the accuracy of net-load forecast [32]. Improved probabilistic net-load forecast would enable more efficient and reliable operation of the distribution systems, thereby realizing the operational security and resiliency of the US power grid, a critical energy infrastructure. Moreover, increased economic value potential of BTM solar resources, driven by a lower cost of forecast error due to accurate net-load prediction and associated requirement of less reserves requirements, would help meet the SunShot 2030 goal of 50% reduction in levelized cost of (solar) energy[1]. Successful completion of the proposed effort would enable large-scale adoption of BTM solar PVs on the distribution networks, thereby helping achieve the Renewable Portfolio Standards (RPS) mandates in various US states.

[1] US DOE SunShot 2030, (online) `https://www.energy.gov/eere/solar/sunshot-initiative`, 2017

2.3 Tasks Outline

Following are the list of major tasks broken down into each budget period.

Budget Period 1 (BP1)

Task 1.0: Stakeholder Engagement

Task Summary: The objectives of this task are to organize regular (quarterly/bi-annual) meetings with relevant stakeholders, including utilities and vendors; and generate technical reports and/or peer-reviewed publications to disseminate the technical findings.

Task 2.0: VRN3P Model Development

Task Summary: The objectives of this task are to design and implement a novel deep variational recurrent neural network (RNN) architecture for net-load forecasts, by integrating a variational autoencoder (VAE) with a recurrent neural network; and provide theoretical assessment of the forecast accuracy.

Subtask 2.1: Integrated Design of RNN with VAE

Subtask Summary: The main objective of this subtask is to design and implement the novel deep VRNN architecture by combining a VAE and an RNN.

Subtask 2.2: Theoretical Assessment of Tolerable Uncertainty

Subtask Summary: The main objective of this subtask is to perform theoretical assessment of the integrated VAE and RNN framework to establish an upper bound on the tolerable uncertainties in the input dataset (e.g., sensor measurements and meters data).

Task 3.0: VRN3P Model Validation

Task Summary: The main objectives of this task are to curate anonymized real-world AMI records, metered/sub-metered load and solar data received from the utility partner; develop multiple scenarios representing various solar penetration (for instance, 20%, 30%, and 50% in annual energy); and perform validation of the proposed VRN3P framework on at least 1 scenario (20% BTM solar in annual energy) to demonstrate 2% MAPE in forecast error.

Subtask 3.1: Data Curation

Subtask Summary: The main objective of this subtask is to curate the real-world, anonymized dataset received from the utility partner for training and validation.

Subtask 3.2: Test Scenarios Development

Subtask Summary: The main objective of this subtask is to construct multiple test scenarios for validation, representing different high-solar penetration levels (20%, 30%, 50% in annual energy) and seasonal/daily end-use variations.

Subtask 3.3: Model Performance Evaluation

Subtask Summary: The main objective of this subtask is to perform a validation of the VRN3P model on 1 test scenario with 20% BTM solar (in annual energy) and demonstrate a net-load forecast accuracy of 2% hourly MAPE.

Task 4.0: VRN3P Tool Specification

Task Summary: The main objectives of this task are to specify the inputs, outputs, and key functionalities of a tool that performs VRN3P forecasting and evaluate the feasibility of hosting it on the GridAPPS-D platform.

Budget Period 2 (BP2)

Task 5.0: VRN3P Tool Specification

Task Summary: The main objective of this task is stakeholder engagement and dissemination of project findings, including via organized workshops and technical panel sessions, and publication of technical reports and peer-reviewed papers detailing the design description and results.

Task 6.0: VRN3P Model Extension

Task Summary: The main objective of this task is to extend the deep variational RNN architecture developed in Budget Period 1 with the design and implementation of transfer learning models. The main outcome of this task is a deep transferable VRN3P framework with 10x training efficiency (relative to baseline VRN3P model developed in Budget Period 1) while achieving target 2% hourly MAPE forecast accuracy.

Task 7.0: VRN3P Extended Performance Evaluation

Task Summary: The main objectives of this task are to a) perform a comprehensive validation of the transferable VRN3P framework on the test-case scenarios constructed in Budget Period 1, representing higher BTM solar penetration (e.g., 30%, 50% in annual energy), with demonstrated forecast accuracy of 2% hourly MAPE; and b) assess the economic benefits of VRN3P leveraging PNNL GRAF-Plan tool, relative to the utility partner's baseline scenario.

Task 8.0: VRN3P Tool Development

Task Summary: The main objective of this task is to develop an open-source validated VRN3P tool for net-load forecasting, based on the list of the inputs, outputs, and key functionalities specified in Budget Period 1.

2.4 Milestones

The following is a list of major milestones and go/no-go decision point(s).

2.4 Milestones

#	Description	Date
1.3.1	Sample dataset curation, with a success metric of a minimum of 1 week of time-series data from each of 4+ different seasons of the year.	06/30/21
1.2.1	Integration of VAE and RNN architectures, with a success metric of 2% hourly MAPE on the sample dataset	09/30/21
1.2.2	Theoretical assessment of tolerable uncertainty bounds, with a success metric of at least 80% of the empirically identified tolerable uncertainty bounds commensurate with 2% hourly MAPE target	12/31/21
1.3.2	Test-case scenarios development, with a success value of a minimum of 1 test scenario with at least 20% BTM solar penetration.	12/31/21
1.1.1	Peer-reviewed publication, with a success value of at least 1 publication submitted to peer-reviewed journal/conference	03/31/22
GNG-1A	Empirical validation of the VRN3P model on 1 selected test-case with 20% BTM solar (in annual energy) and a demonstrable net-load forecast accuracy of 2% hourly MAPE, averaged over all valid hours during a minimum of 6 months, measured at the substation.	03/31/22
1.3.3	Test-case scenarios development, with a success value of a minimum of 2 additional test-cases with at least 30-50% BTM solar penetration.	06/30/22
2.7.1	Economic baseline selection, with at least 1 baseline scenario for economic assessment	09/30/22
2.6.1	Preliminary integration of deep transfer learning network with VRN3P, with a success value of 2% MAPE on at least 2 test-cases with 20-30% BTM solar, and at least 10x training efficiency over the non-transferable design in BP1	12/31/22
2.8.1	Initial design of VRN3P user tool (version-“beta”), with a success value of 1 open-source tool with limited functionalities, e.g., multi-data entry and visualization	03/31/23
2.8.2	Final design of open-source VRN3P (version-“1.0”), with a success value of 1 open-source tool released with full functionalities	06/30/23
2.5.1	Peer-reviewed publications, with a success value of at least 2 publications submitted to peer-reviewed journals and/or conferences	06/30/23
EOP-A	Extended (transferable) VRN3P validation, with a success value of 2% hourly MAPE on at least 3 test-cases with 20-50% BTM solar, and at least 10x training efficiency gain over non-transferable VRN3P	09/30/23
EOP-B	Economic benefit estimation of enhanced forecast, with a success value of 5% reduction in operational cost due to forecast error, averaged over 6+ months in a typical year	09/30/23
EOP-C	Workshop organization, with a success value of 1 workshop organized	09/30/23

3 Project Results and Discussion

The next sub-section summarizes a high-level, quantitative comparison of anticipated project outcomes and milestones against realized results, broken down into tasks. The detailed results and discussions are provided in the following sub-section.

3.1 Summary Outcomes and Realized Results

Task 1 - Stakeholder Engagement

#	Description	Date
1.1.1	Peer-reviewed publication, with a success value of at least 1 publication submitted to peer-reviewed journal/conference	03/31/22
	Realized Results: Prepared a detailed technical manuscript, reporting on the novel deep probabilistic net-load forecasting algorithm, for IEEE PES journal and uploaded it to arXiv for public access. A copy of the paper was sent to DOE.	

Task 2 - VRN3P Model Development

#	Description	Date
1.2.1	Integration of VAE and RNN architectures, with a success metric of 2% hourly MAPE on the sample dataset	09/30/21
	Realized Results: Completed preliminary design and implementation of an integrated VAE and RNN architecture, trained on the sample NEEA dataset, and demonstrated a forecast performance improvement of >30% over the state-of-art LSTM-based models, as well as the probabilistic convolution-LSTM model developed in BP1Q1.	
1.2.2	Theoretical assessment of tolerable uncertainty bounds, with a success metric of at least 80% of the empirically identified tolerable uncertainty bounds commensurate with 2% hourly MAPE target	12/31/21
	Realized Results: Developed a data-driven nonlinear sensitivity based approach to theoretically quantify the tolerable uncertainty bounds on the input data, for acceptable forecast performance. Validated that the error statistics generated from the theoretical calculation matches the model forecast error with $> 88\%$ closeness.	

Task 3 - VRN3P Model Validation

#	Description	Date
1.3.1	Sample dataset curation, with a success metric of a minimum of 1 week of time-series data from each of 4+ different seasons of the year.	06/30/21
	Realized Results: Over 1 year worth of 1-min/15-min sampled anonymized real-world data comprising of sub-metered solar and other appliances' energy consumption was received and curated by project team. Data represents hundreds of residential and commercial customers spread across the states of WA, OR, and ID, along with associated weather data, collected under NEEA EULR effort.	
1.3.2	Test-case scenarios development, with a success value of a minimum of 1 test scenario with at least 20% BTM solar penetration.	12/31/21
	Realized Results: Constructed a 12.47 kV test-case feeder (R1-12.47-2 from the Modern Grid Initiative), with the characteristics of a primarily residential feeder in climate zone 1 (includes the Pacific Northwest), representative of suburban Portland, OR. The test case feeder was modified to accept 251 GridLAB-D player files each of which is a unique net-load residential time series from PGE.	
1.3.3	Test-case scenarios development, with a success value of a minimum of 2 additional test-cases with at least 30-50% BTM solar penetration.	06/30/22
	Realized Results: Synthesized 3 test-case feeders with varying penetration of BTM solar (in terms of annual energy) at, specifically, 20%, 30%, and 50% . The anonymized 15-minute sampled AMI data from 95,000+ PGE customers were used to generate the test-cases, with varying solar penetration levels.	

Task 4 - VRN3P Tool Specification

Brief Summary of Actual Activities:

- Preliminary outline of the key input, output, and functionalities of the proposed VRN3P application was developed. GridAPPS-D™ was considered as a platform to host the application, for it being an open-source, open-structure, standards-based platform, with a web-based user-interface for visualization.

Go/No-Go Decision Point

The go/no-go milestone (GNG-1A) was successfully achieved with the following realized outcome.

#	Description	Date
GNG-1A	Empirical validation of the VRN3P model on 1 selected test-case with 20% BTM solar (in annual energy) and a demonstrable net-load forecast accuracy of 2% hourly MAPE, averaged over all valid hours during a minimum of 6 months, measured at the substation.	03/31/22
	Realized Results: The project team completed successful design of a novel probabilistic net-load forecasting architecture, comprising of a VAE and an RNN, which demonstrated 30% improvement in forecast performance (as per metrics such as PBB and CRPS), 60% improvement in training time, and consumed 44% less memory, when compared with conventional baseline models (e.g., probabilistic convolutional-LSTM (P-CLSTM).	

Task 5 - Stakeholder Outreach

#	Description	Date
2.5.1	Peer-reviewed publications, with a success value of at least 2 publications submitted to peer-reviewed journals and/or conferences	06/30/23
	Realized Results: Two papers were submitted (and later accepted) for publication at the IEEE PES conferences: ISGT 2024, and PESGM 2024. Another paper, jointly supported by this project, was accepted for publication at the IEEE PES Grid Edge 2025	

Task 6 - VRN3P Model Extension

#	Description	Date
2.6.1	Preliminary integration of deep transfer learning network with VRN3P, with a success value of 2% MAPE on at least 2 test-cases with 20-30% BTM solar, and at least 10x training efficiency over the non-transferable design in BP1	12/31/22
	Realized Results: The project team tested the performance of transfer-VRN3P model and observed 8.33× speed-up in training time, while achieving acceptable forecast performance of 2.24% hourly MAPE (averaged over a year), with a model trained on 20% BTM solar being transferred over to 30% BTM solar test-case	

Task 7 - VRN3P Extended Performance Evaluation

#	Description	Date
2.7.1	Economic baseline selection, with at least 1 baseline scenario for economic assessment	09/30/22
	Realized Results: Set up a preliminary framework for economic impact evaluation of day-ahead net-load forecast on the utility partner's (PGE's) system, emulating its participation in CAISO Energy Imbalance Market (Real Time Market) at the PGE trading node.	

Task 8 - VRN3P Tool Development

#	Description	Date
2.8.1	Initial design of VRN3P user tool (version-“beta”), with a success value of 1 open-source tool with limited functionalities, e.g., multi-data entry and visualization	03/31/23
	Realized Results: The project team completed the design of the VRN3P GridAPPS-D™ application modules for basic functionalities, along with a web-based user-interface	
2.8.2	Final design of open-source VRN3P (version-“1.0”), with a success value of 1 open-source tool released with full functionalities	06/30/23
	Realized Results: A web-based user-interface was developed and released under open-source license on GitHub (link: `https://github.com/pnnl/Forte`) to help disseminate the outcome of the VRN3P framework to the end users via an easy-to-use visual and interactive interface.	

Final Outcomes

#	Description	Date
EOP-A	Extended (transferable) VRN3P validation, with a success value of 2% hourly MAPE on at least 3 test-cases with 20-50% BTM solar, and at least 10x training efficiency gain over non-transferable VRN3P	09/30/23
	Realized Results: The project team designed a probabilistic net-load forecasting architecture, comprising of a VAE and an RNN, demonstrating 30% improvement in forecast performance, 60% improvement in training time, while consuming 44% less memory, when compared with conventional baseline models. Tested on the 20% BTM solar test-case, the developed VRN3P model achieved $<$2% hourly MAPE averaged over the year. Transfer learning extended VRN3P model demonstrated $8.33\times$ speed-up in training, while achieving forecast performance of 2.24% hourly MAPE on the 30% BTM solar case. Tested on the Solar Forecast Arbiter, the model achieved high CRPSS +0.3 and +0.08, compared to a given reference, in two locations with different solar penetration in TX (19%) and GA (63%), respectively.	
EOP-B	Economic benefit estimation of enhanced forecast, with a success value of 5% reduction in operational cost due to forecast error, averaged over 6+ months in a typical year	09/30/23
	Realized Results: Since most utilities (including PGE) do not use feeder-level net-load forecasts in operational decisions, the project team chose to measure VRN3P financial impact at the balancing authority (BA) level emulating participation in CAISO Energy Imbalance Market (Real Time Market) at the PGE trading node. Using PNNL GRAF-Plan tool, significant (60-95%) reduction in day-ahead reserve requirement was estimated, when compared to CAISO 2020 forecast benchmark.	
EOP-C	Workshop organization, with a success value of 1 workshop organized	09/30/23
	Realized Results: A panel session on net-load forecasting was organized at an IEEE PES venue: ISGT 2022. The work was presented at the SETO AI/ML Workshop 2023.	

3.2 Detailed Results and Discussions

3.2.1 Data Overview

NEEA/EULR Dataset: PGE shared with the project team dataset containing sub-metered solar and electric load profiles, at 15-min resolution, for hundreds of residential and commercial customers over 1 year (2018-2020), collected in partnership with NEEA as part of the EULR project. The dataset currently covers 28 months, starting in Sep 2018 till Dec 2020. It consists of power usage by 3000 circuits across 198 buildings. Other measurements such as humidity, interior and exterior temperatures and weather data are also provided. The data covers power consumption across several states and climatic zones. Even though the data does not specify the exact location (latitude and longitude), one may infer proximity of the test sites from its corresponding weather station id. The net-load (NL) of site is defined as the measure power at the 'Mains' circuit of that site. Additionally, sites with solar installations have a dedicated 'Solar PV' circuit, that measures the solar photovoltaic power generated on-site. For some sites, there exists multiple temperature measurements, both exterior and interior. For our purposes, we take the mean exterior temperature (if available) over all exterior temperature transmitters at each site. As an example, a weather station with ID: 720638-224 in Oregon has a total of 19 sites associated with it, out of which 7 have solar installations. The 7 solar customers consume an annual aggregated demand of about 70 MWh, while producing an aggregated solar generation of about 25 MWh – which amounts to roughly 36% of BTM solar penetration.

PGE Data and Associated Challenges: Furthermore, load and solar generation time series data from AMI meters within the PGE territory were also shared with the project team. PGE has two types of programs for customers with rooftop solar: (1) Solar net-meter, where one meter records net-load (capped at zero) and another meter records net-generation in an interval, and (2) Solar Purchase Option (SPO), which provides signals for net-load, net-generation, and total interval generation. The PGE dataset is comprised of three types of customers: (1) customers without solar, (2) solar customers with a net-metering agreement, and (3) solar customers with SPO and net-metering agreement. The dataset provided by PGE contained net consumption time series for 24,438 customers inclusive of the three customer types. Across sub-types, 63% are without solar, 36% are net-meter only, and 1.6% are net-meter and SPO. The break-out of customer counts and usage/generation is shown in Table 1. Fig. 1 (Top) illustrates the anonymized net generation (excess generation exported to the grid) and net consumption (amount billed to the customer) for one customer over 24 hours.

Table 1: PGE residential customer counts and annual energy.

Utility Dataset	Count	Annual Sum
No Solar	15,289	140 GWh
Net-meter Net Generation	7,862	28 GWh
Net-meter Net Consumption	8,766	73 GWh
SPO Net Generation	369	1.4 GWh
SPO Net Consumption	383	5 GWh
SPO Total Generation	383	2.5 GWh

Net-load forecasting accuracy is a function of the impact of interval generation on the native load. Solar generation measurements are only available for 383 customers and many of these customers have sized their rooftop PV systems to meet their annual demand; this translates to a depressed

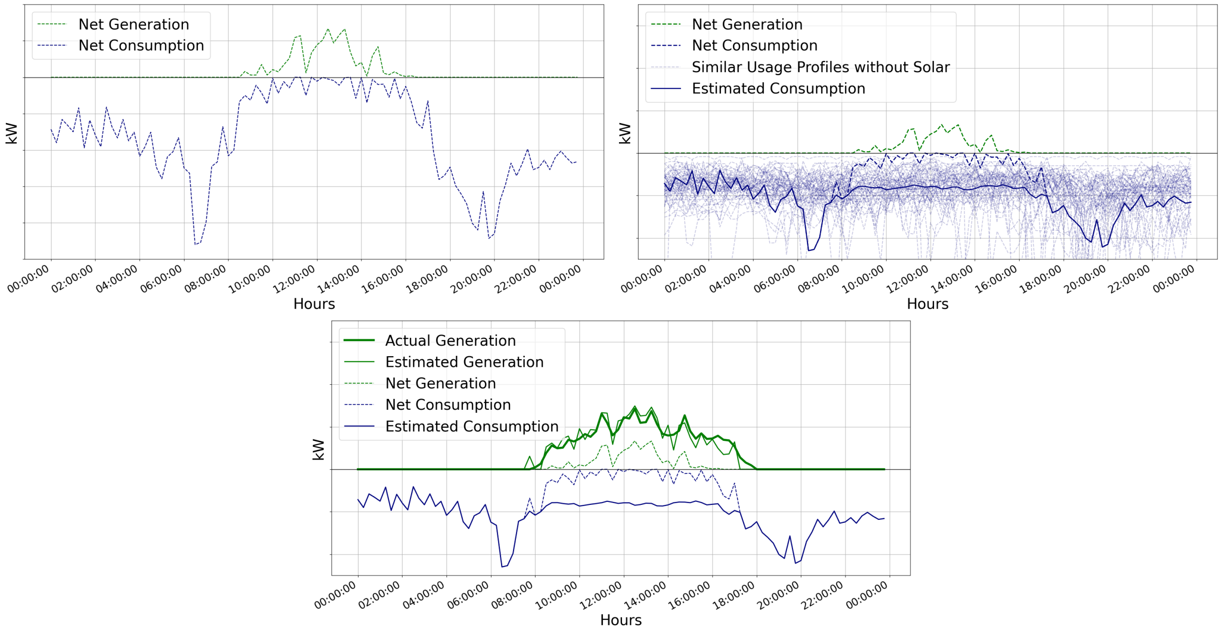


Figure 1: (Top) Net generation (green) and the net consumption (blue) of one customer. (Middle) Interval consumption of 75 similar non-solar customers. Light blue lines are individual non-solar customer usage. (Bottom) Extra generation identified by this approach effectively recaptures interval generation. Exact kW values on y-axis are removed to retain anonymity.

percentage of household-level solar penetration by annual energy self-produced. Based on net generation and net consumption of all 383 SPO customers, the solar penetration is $\sim 28\%$. In order to reach higher penetration levels upwards of 50% to test the robustness of net-load forecast algorithms, this methodology must reconstruct the additional BTM generation of net-meter solar customers. These customer profiles are then recombined to produce synthetic feeders with high BTM solar generation.

3.2.2 Estimating BTM Solar for Scenario Synthesis

Solar generation depends on reliable solar irradiance which is influenced by cloud coverage. While cloud coverage on varied-condition days changes across the extent of a balancing authority (PGE is $\sim$60 miles across), an average-sized cloud can completely eclipse a distribution feeder; all BTM generation in the feeder will experience similar reduced output from the passing cloud. We reconstruct the average BTM generation on days with clear weather by looking to the native usage of similar non-solar customers on overcast days. Averaging the profiles of customers with similar usage reduces downstream variance in marginal PV generation estimation which could arise from variations in PV installation status and equipment degradation. The goal is to estimate the average marginal native load for the target customer under clearsky conditions.

Usage profiles from non-solar customers are deemed similarity candidates to those of SPO customers on days when consumption is unlikely be impacted by BTM solar generation or when GHI is very low. Weather will be the dominant indicator of PV production; on rainy days PV output can be reduced by 40% and on overcast days by 45% [14]. Interval cloud cover condition is used as a proxy for GHI with condition observations taken from the Portland International Airport [18].

Weather conditions and their similarity weighting (denoted by S) is shown in Table 2. Weights are assigned to a simplified weather condition, such as $S = 1$ (rainy), $S = 0.1$ (cloudy), and $S = 0.01$ (fair) to logarithmically preference the similarity of non-solar interval consumption during rainy intervals over others. The specific weight values may be tuned for improved accuracy.

Table 2: Weather conditions observed at the Portland Intl. Airport.

Weather Condition	Group	Similarity Weight (S)
Heavy Rain, Rain	Rainy	1
Light Rain, (Mostly/Partly) Cloudy, (Shallow/Patches of) Fog, Mist, Haze, Light Drizzle, Wintry Mix, Smoke, Heavy Thunder-Storm	Cloudy	0.1
Fair, Light Snow, Windy, Thunder	Fair	0.001

We leverage non-solar customer profiles to reconstruct native SPO load on overcast days [6], following a similar approach as in [26]. There are no direct GHI measurements to reconstruct generation; in its place, recorded weather conditions (rainy/cloudy/fair) at the nearest weather station are used as proxy for sky clearness (Table 2). Testing of the algorithm is done with SPO customers supplemented with usage profiles from non-solar customers. The algorithm to complete this task for each SPO customer has three steps (see [6] for details):

Step 1: *Calculate Daily Averaged Similarity Weight ($\overline{S}_d$).* Considering only the daytime intervals (i.e., sunrise to sunset) containing a total of m time-intervals $\{t_1, t_2, \ldots, t_m\}$, the daily averaged similarity weight for the feeder region is obtained as:

$$\overline{S}_d = \frac{S_{t_1} + S_{t_2} + \cdots + S_{t_m}}{m} \qquad \text{(for each day } d\text{)} \tag{1}$$

where S_t denotes the similarity weight at interval t .

Step 2: *Select Similar Non-Solar Load Profiles.* For every solar customer i, we define the interval difference between their net usage profile ($u_{i,t}$) and the usage profile ($u_{j,t}$) of every non-solar customer j as follows:

$$\Delta_{i,j,t} := \left|u_{i,t} - u_{j,t}\right|, \quad \forall i : \text{\{solar customers\}},\ \forall j : \text{\{non-solar customers\}},\ \forall t : \text{\{time-intervals\}} \tag{2}$$

Considering the m day-time intervals $\{t_1, t_2, \ldots, t_m\}$, the daily usage difference between the i-th solar customer and the j-th non-solar customer is obtained as:

$$\Delta_{i,j,d} = \Delta_{i,j,t_1} + \Delta_{i,j,t_2} + \cdots + \Delta_{i,j,t_m} \qquad \forall d : \text{\{days\}} \tag{3}$$

Next, the daily usage differences are weighted by the daily averaged similarity weights ($\overline{S}_d$) and summed over a full year to obtain the weighted annual usage difference between the i-th solar customer and the j-th non-solar customer as follows:

$$\Delta_{i,j} = \sum_{d \in \text{\{days in year\}}} \left(\overline{S}_d \cdot \Delta_{i,j,d}\right) \tag{4}$$

Note that small Δ_{ij} suggests that the usage (net consumption) profile of the j-th non-solar customer is similar to the i-th solar customer. The empirical distribution of the Δ_{ij} values are used to identify the non-solar similar customers as:

$$\forall i : \quad \mathcal{N}_i := \left\{ j \,\middle|\, \Delta_{ij} \leq \text{med}\left(\Delta_{ij}\right) - \sigma\left(\Delta_{ij}\right) \right\}, \tag{5}$$

where med$(\cdot)$ and $\sigma(\cdot)$ denote median and standard deviation, respectively. Only the customer profiles that are in the tail of the distribution — at least one standard deviation below the median — are considered as similar.

Step 3: *Correct Load and Generation on Fair Days.* For each solar customer i, we have access to the net-metered values:

$$u_{i,t} \leq 0 : \text{net consumption}\,, \quad v_{i,t} \geq 0 : \text{ net generation}\,.$$

The goal is to estimate the total generation ($\hat{v}_{i,t}$), by adjusting the net-generation values with the help of the estimated total consumption ($\hat{u}_{i,t}$), calculated as follows:

$$\hat{u}_{i,t} = \frac{1}{|\mathscr{N}_i|} \sum_{j \in \mathscr{N}_i} u_{j,t}\,,$$

where $|\mathscr{N}_i|$ denotes the number of similar non-solar customers. Finally, the total generation ($\hat{v}_{i,t}$) is estimated by adding a non-negative correction ($w_{i,t} \geq 0$) to its net generation ($v_{i,t}$) :

$$\hat{v}_{i,t} = v_{i,t} + w_{i,t}\,, \qquad \text{where } w_{i,t} := \max(0, u_{i,t} - \hat{u}_{i,t}) \tag{6}$$

The resulting interval consumptions generated by the algorithms are shown in Fig. 1. Validation of the BTM solar generation reconstruction was done with SPO customers using their actual generation meter readings as comparison. Across these customers, the net generation series only accounts for 52% of the total annual generation. Reconstitution of the interval generation using the approach in this work increases the representation of BTM generation to 75% of total annual generation. The error in the annual generation can be seen in Fig. 2, where the percent error for each customer is the annual actual generation minus net (or estimated) generation divided by actual. The error in the net generation is 48%, with a standard deviation of 15%. After application of this approach, the error in the estimated generation is 26% with a standard deviation of 11%.

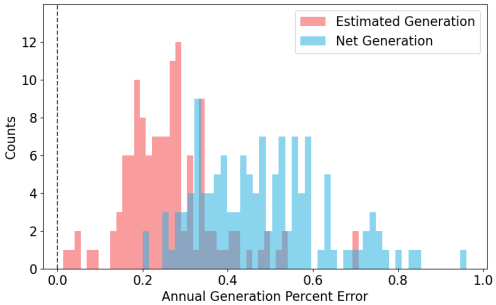


Figure 2: Customer-level percent error in annual generation for estimated (red) and net-meter (blue) values.

The mean absolute percent error (MAPE) in monthly energy averaged across all SPO customers is shown in Fig. 3. Where Fig. 2 showed the annual error, this dives into the seasonal dependence. We see that this reconstruction approach improves the estimate of BTM solar generation in the summer months most. This approach selected non-solar customer usage profiles based on the integrated usage difference on overcast or rainy days; this allows us to leverage the same usage profiles to reconstruct the native demand on sunny days.

The purpose of calculating the total generation and total consumption for each SPO customer is to build scenarios of BTM solar penetration from aggregations of different customers. The methodology was introduced with SPO customers and can be applied to any customer with a net-meter

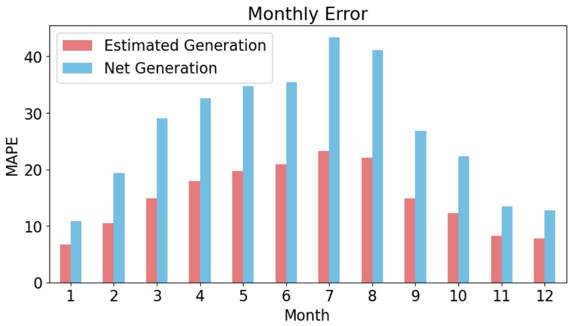


Figure 3: Averaged across validation customers, the monthly MAPE is lower for the proposed approach to estimate BTM generation compared with the metered net generation.

profile. The BTM generation reconstruction algorithm has been applied to solar customers without a meter for total generation ($\sim 36\%$ of customers in PGE, Table 1). Recalling that many solar customers size their systems to have no excess generation, the methodology described in this work will allow users to accurately represent BTM solar penetration and re-aggregate to artificial levels for testing of net-load forecasts. A demonstration of this re-aggregation to produce high penetrations of BTM solar service territories is shown in Fig. 4 for three scenarios. These figures are on the same vertical scale to emphasize the shift in net consumption (green, positive) to net generation (blue, negative).

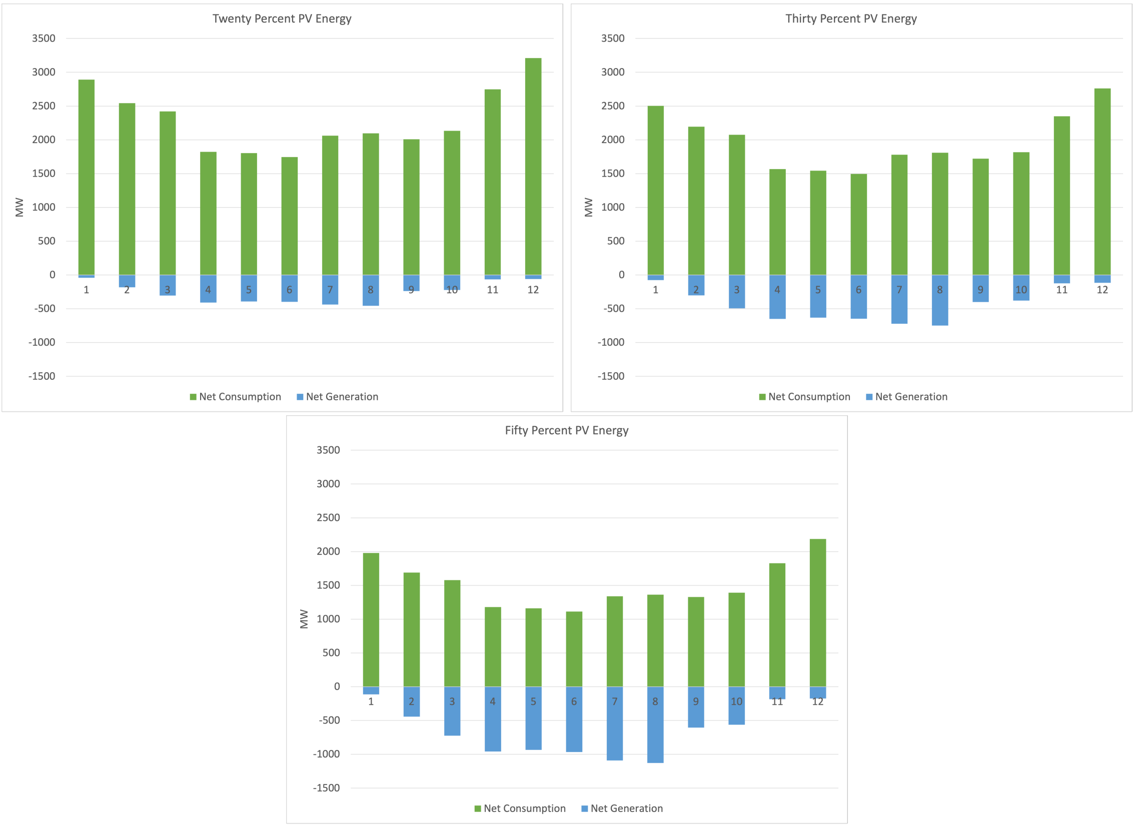


Figure 4: Monthly consumption and generation for three BTM solar levels: 20% , 30% , 50% (right).

3.2.3 Test Case Feeder Topologies

The project team constructed a test case feeder to account for the high-solar scenarios. This first test case was built from the Modern Grid Initiative feeder taxonomy number R1-12.47-2 [36]. This feeder was selected to mimic the characteristics of a feeder in climate zone 1 (which includes the Pacific Northwest) with primarily residential loads, representative of suburban Portland, OR in the PGE service territory. This feeder, shown in Figure 5 (with details in Table 3), has a 12.47 kV distribution primary, which is a typical voltage level for residential feeders. The test case feeder was modified to accept GridLAB-D player files on each triplex node that contains a triplex load parameter. Every GridLAB-D player file simulates a collection of residential loads. An example of the triplex node simulating a house is shown in Figure 6, where the pentagon symbolizes a house. Each player file is a unique net load residential time series from PGE, meaning that the replacement of simulated load in the test case is one-to-one with real PGE customer load data.

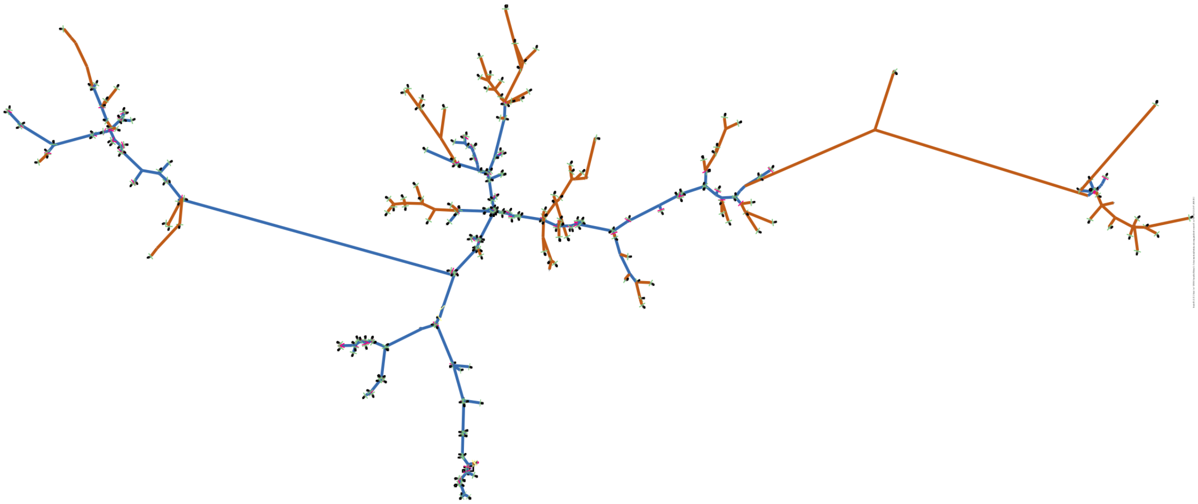


Figure 5: Feeder configuration for R1-12.47-2.

Table 3: Feeder characteristics for R1-12.47-2

Nodes	337
Voltage (kV)	12.47
Load (kW)	2,830
Voltage Regulators	0
Reclosers	0
Residential Transformers	227
Commercial Transformers	13
Industrial Transformers	0
Agricultural Transformers	0

The selected taxonomy feeder R1-12.47.2 contains 251 triplex nodes that simulate residential loads (or triplex_node objects with a power_12 parameter in GridLAB-D terminology). Distribu-

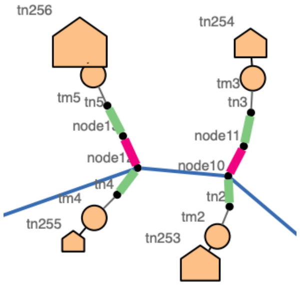


Figure 6: Triplex nodes in R1-12.47-2 (e.g., tn256) were adapted to accept load time series. The meter upstream (tm5) records the load at the house.

tions of the load values of the 296 load player files — containing 15-min sampled load measurements from net-metered, non-SPO customers for the year 2020 — revealed the inadequacy of individual customer loads to match the nominal load expected at the triplex nodes of the taxonomy feeder, and, therefore, the need to adopt an aggregation procedure. The goal here was to establish (if possible) a relationship between the number of metered customers and the maximum value of the corresponding aggregated load shape. After removing two outlier player files with atypical high load, a sufficiently linear relationship (with an R^2 coefficient of 0.94) was found between the number of metered customers and the maximum value of the corresponding aggregated load shape. The resulting linear model was used to aggregate the load profiles for the triplex nodes in the test feeder.

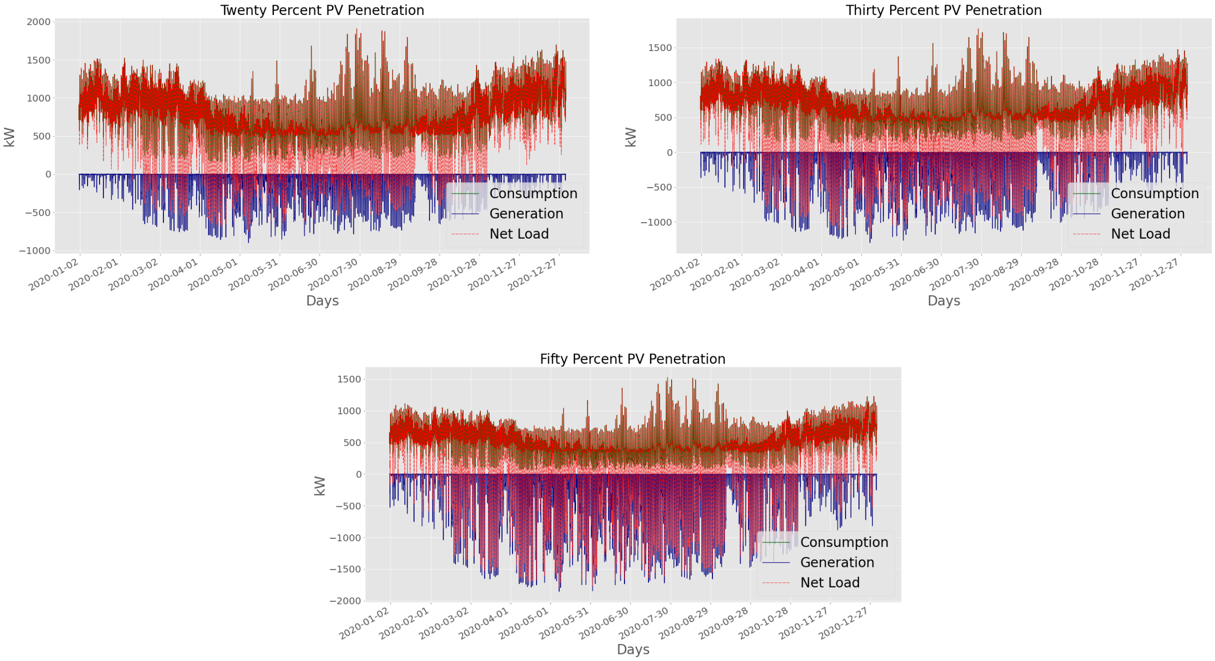


Figure 7: Interval power consumption, generation, and net consumption for the three synthesized BTM solar test-case scenarios: 20%, 30%, and 50%, for the full year.

The resulting interval time series for the three test scenarios are shown in Figure 7, with varying BTM solar penetration of 20%, 30%, and 50%. Net consumption is shown in green (positive), net generation in blue (negative), and net-load in red.

3.2.4 AI/ML Model: Proposed Architecture

One of the most commonly used machine-learning models for time-series forecast is the recurrent neural network (RNN), including long short-term memory (LSTM) network. The variational autoencoder-based RNN (VAE-RNN) architecture for probabilistic net-load forecasting, developed in BP1Q2, was demonstrated to provide superior performance compared to the baseline models in terms of forecast accuracy and training efficiency. Details of the proposed model and performance analysis is documented in [37].

Training Loss Functions and Performance Metrics: *Mean squared error* (MSE) and *mean absolute error* (MAE) remain two of the most commonly used training loss functions for deterministic forecast models. However, for probabilistic models, where the output is a distribution, rather than a single value (yet, the observation is a single value), one may use loss functions such as mean *negative log-likelihood* (NegLL) or mean *continuous ranked probability score* (CRPS), as described below. The NegLL loss is computed as follows:

$$NegLL = \frac{1}{T}\sum_{t=1}^{T}\left(-\log(P(y_{obs}|\theta_{pred}(t)))\right) \tag{7}$$

which is the likelihood that the observed value of target variable ($y_{obs}(t)$) was generated by the predicted probability distribution, which is parameterized by $\theta_{pred}(t)$. For normally distributed target variable, the output distribution is parameterized by the predicted mean ($\mu_{pred}(t)$) and standard deviation ($\sigma_{pred}(t)$). Another alternative loss function for probabilistic outputs is the continuous ranked probability score (CRPS). Given a predicted cumulative density function (CDF), $F_{pred}(y)$, and the CRPS is defined as:

$$crps(F_{pred}, y_{obs}) = \int_{-\infty}^{y_{obs}} \left(F_{pred}(y)\right)^2 dy + \int_{y_{obs}}^{\infty} \left(1 - F_{pred}(y)\right)^2 dy \tag{8a}$$

As a loss function, we compute the average CRPS across the entire training dataset as

$$CRPS = \frac{1}{T}\sum_{t=1}^{T} crps(F_{pred}(t), y_{obs}(t)) \tag{9}$$

Commonly used performance metrics for deterministic forecasts include MAE, and the mean absolute percentage error (MAPE). However, MAPE explodes in value when the observed solar or net-load value is close to zero, which is the expected under extreme high solar penetration. Alternative metrics for deterministic forecast includes the *normalized MAE* ($normMAE$). For the probabilistic forecast, we use the metric CRPS (9), as described before. Additionally, we also report the “probability between bounds” (PBB):

$$PBB = P\left(\mu_{pred} - \sigma_{pred} < y_{obs} < \mu_{pred} + \sigma_{pred}\right) . \tag{10}$$

Proposed VAE-RNN Architecture: In the proposed novel architecture, we utilize the latent representation from an autoencoder after applying kernel-based Perron-Frobenius (kPF) operator, as introduced in [17], along with a probabilistic long-short-term-memory (LSTM) network for net-load forecasting. First, we briefly recap a few fundamental concepts associated with generative modeling, kPF operator and probabilistic LSTM.

Given independent and identically distributed (i.i.d.) samples X with an unknown distribution P_X, a generative model seeks to find a parametric distribution that closely resembles P_X. A popular example for a generative model is the variational autoencoder (VAE) wherein the data generating distribution is learned via latent variables. In other words, we assume that there is some variable $Z \in \mathscr{Z}$ associated with the observed data $X \in \mathscr{X}$ that follows a known distribution P_Z (also referred to as the prior in generative models). Then we can learn a mapping $f : \mathscr{Z} \rightarrow \mathscr{X}$ such that the distribution after transformation, denoted by $P_{f(Z)}$, aligns well with the data generating distribution P_X. For VAEs, f is parameterized by deep neural networks and f is optimized using Maximum Likelihood Estimation (MLE), by minimizing the difference between $P_{f(Z)}$ and P_X (the operator which learns to minimize the distance between $P_{f(Z)}$ and P_X is called a forward operator, as introduced in [17]). To learn the forward operator, VAEs typically use an approximate posterior $q_{Z|X}$ that may sometimes fail to align with the prior ([21]). This motivates the necessity of using kPF operator based training process for training AE model, instead of training a VAE.

Perron-Frobenius (PF) operator [30] is a infinite dimensional linear operator, which can replace the previously mentioned forward operator such that $P_X = PF(P_Z)$. PF is infinite dimensional and there exists many finite dimensional estimation for PF. We have selected Kernel-embedded Perron-Frobenius (kPF) method to calculate finite dimensional estimation as introduced in [22]. Our proposed kPF-Autoencoder (kPF-AE) architecture consists of a one-dimensional convolutional autoencoder, which finds a latent space (encoding space) to represent a historically stacked input space. Subsequently, the kPF operator is applied on the resulting latent space using Algorithm 1 to generate the distribution of the latent variables.

Algorithm 1 Sample generation using kPF

```
Input: trained encoder $\mathscr{E}$, training data $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$, neighborhood size $\gamma$
Output: generated $m$ samples $\mathbf{x}^*$
Calculate encoded representation of training sample $\mathbf{X}_e = \Big(\mathscr{E}(\mathbf{x}_1), \mathscr{E}(\mathbf{x}_2), \ldots, \mathscr{E}(\mathbf{x}_n)\Big)$
Calculate $K_{i,j} = k(\mathbf{X}_{e_i}, \mathbf{X}_{e_j}), (i,j)\forall n$, where $k(.,.) = \exp - \left(\frac{\|x-y\|}{2}\right)$
Sample $z$ independently from $\mathscr{Z}$, where $\mathscr{Z}$ is a multi-variable normal distribution with mean $0$ and identity co-variance
Calculate $L_{i,j} = k(z_i, z_j), (i,j)\forall n$
Calculate $K_{inv} = (K + nI)^{-1}$, where $I$ is a $n \times n$ identity matrix
Sample $nz$ independently ($m$ number of ) from $\mathscr{Z}$, where $\mathscr{Z}$ is a multi-variable normal distribution with mean $0$ and identity co-variance
Calculate $nv_{i,j} = k(z_i, nz_j), i\forall n, j\forall m$
Calculate $s = L.K_{inv}.nv$, where . denotes matrix multiplication
Calculate $\mathrm{ind} = \mathrm{argsort}(s)[-\gamma :]$
Calculate $\mathbf{x}^* = \frac{\mathbf{X}_e[\mathrm{ind}]s[\mathrm{ind}]}{\|s[\mathrm{ind}]\|_1}$
```

Bayes by Back-propagation (BBB) [5] is a variational inference scheme for learning the posterior distribution on the weights $\theta \in \mathbb{R}^d$ of a neural network. This posterior distribution is typically taken to be a Gaussian with mean parameter $\mu \in \mathbb{R}^d$ and standard deviation parameter

$\sigma \in \mathbb{R}^d$, denoted $\mathcal{N}(\theta|\mu, \sigma^2)$. Let $\log p(y|\theta, x)$ be the log-likelihood of the model, then the network is trained by minimizing the variational free energy, $\mathcal{L}(\theta) = \mathbb{E}_{q(\theta)}\left[\log \frac{q(\theta)}{p(y|\theta,x)p(\theta)}\right]$. The core of an RNN, f, is a neural network that maps the RNN state s_t at step t, and an input observation x_t to a new RNN state $s_{t+1}, f : (s_t, x_t) \to s_{t+1}$. An RNN can be trained on a sequence of length T by backpropagation through time by unrolling T times into a feed-forward network. Explicitly, we set $s_i = f(s_{i-1}, x_i), \forall i = 1, \ldots, T$. RNN parameters are learnt in much the same way as in a feed-forward neural network. A loss (typically after further layers) is applied to the states $s_{1:T}$ of the RNN, and then backpropagation is used to update the weights of the network. Crucially, the weights at each of the unrolled steps are shared. Thus each weight of the RNN core receives T gradient contributions when the RNN is unrolled for T steps. Applying BBB to RNNs, the weight matrices of the RNN are drawn from a distribution, whose parameters are learned. Therefore, the variational free energy for an RNN on a sequence of length T is, $\mathcal{L}(\theta) = -\mathbb{E}_{q(\theta)}\left[\log p(y_{1:T}|\theta, x_{1:T})\right] + \mathrm{KL}\left[q(\theta)||p(\theta)\right]$.

Our proposed architecture is schematically shown in Figure 8. Our framework takes temporal measurement consisting of historic net-load (y) and weather related measurements (e) such as ambient temperature, relative humidity and wind direction. We will use the temporal stack of these variables to forecast net-load in future time. The main contribution of our framework is the incorporation of the latent representation from the input space, in forecasting net-load. The proposed method also improves the training time (via the use of kPF algorithm) over existing methods such as variational recurrent neural network [7] for probabilistic forecasting.

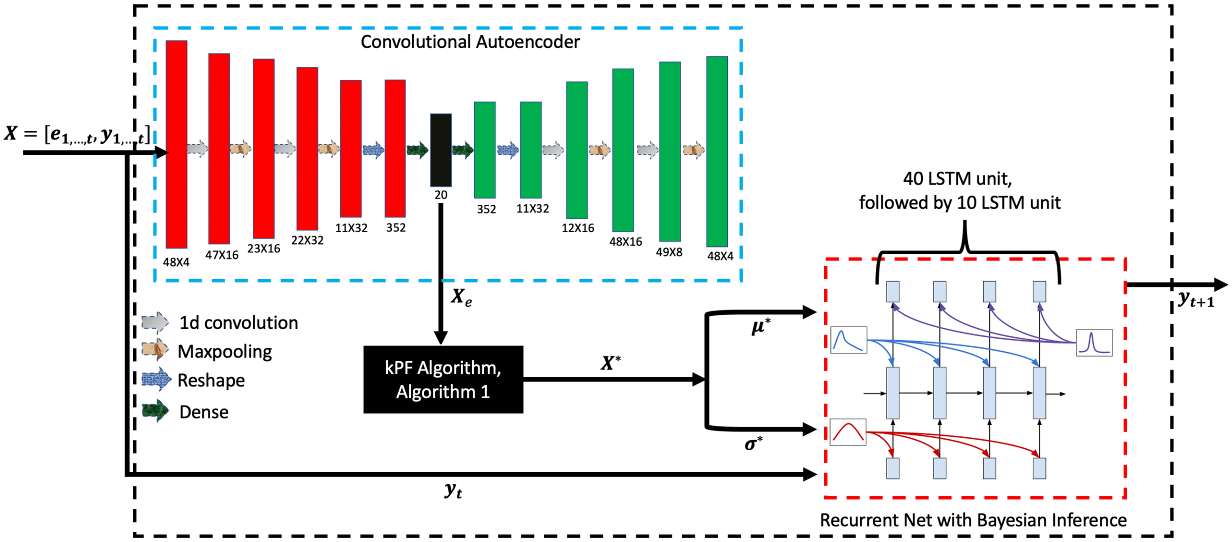

$X = [e_{1,\ldots,t}, y_{1,\ldots,t}]$
Convolutional Autoencoder
48X4
47X16
23X16
22X32
11X32
352
20
352
11X32
12X16
48X16
49X8
48X4
X_e
1d convolution
Maxpooling
Reshape
Dense
kPF Algorithm, Algorithm 1
X^*
μ^*
σ^*
y_t
40 LSTM unit, followed by 10 LSTM unit
y_{t+1}
Recurrent Net with Bayesian Inference

Figure 8: Schematic of our proposed architecture is shown within the black dashed box. The overall framework has primarily three sub-components, convolutional autoencoder (within blue dashed box), kPF Algorithm as shown in Algorithm 1, and a probabilistic LSTM network (within red dashed box).

Performance Analysis: We compared our model performance with existing forecast models, such as convolutional-LSTM (CLSTM) and probabilistic convolutional-LSTM (P-CLSTM) [41], as shown in Figure 9. In addition, the performance of our proposed VRNN model (with sequential training) was compared against conventional VRNN [7]. As shown in the table below, there is a significant

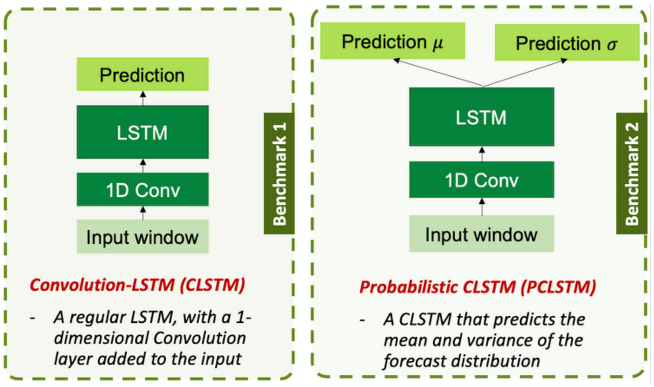
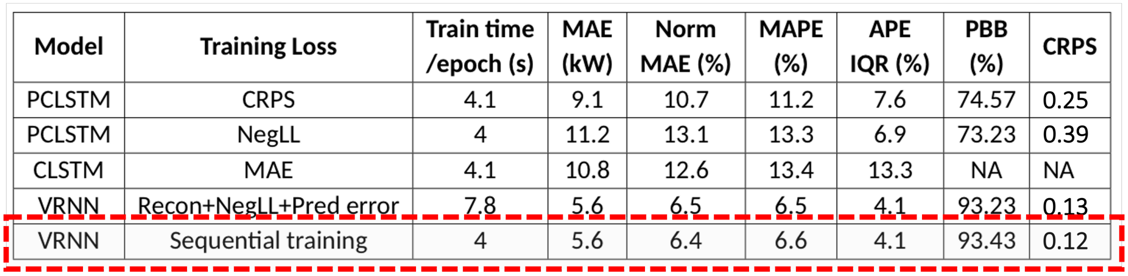

Prediction
LSTM
1D Conv
Input window
Benchmark 1
Convolution-LSTM (CLSTM)
- A regular LSTM, with a 1-dimensional Convolution layer added to the input
Prediction μ
Prediction σ
LSTM
1D Conv
Input window
Benchmark 2
Probabilistic CLSTM (PCLSTM)
- A CLSTM that predicts the mean and variance of the forecast distribution

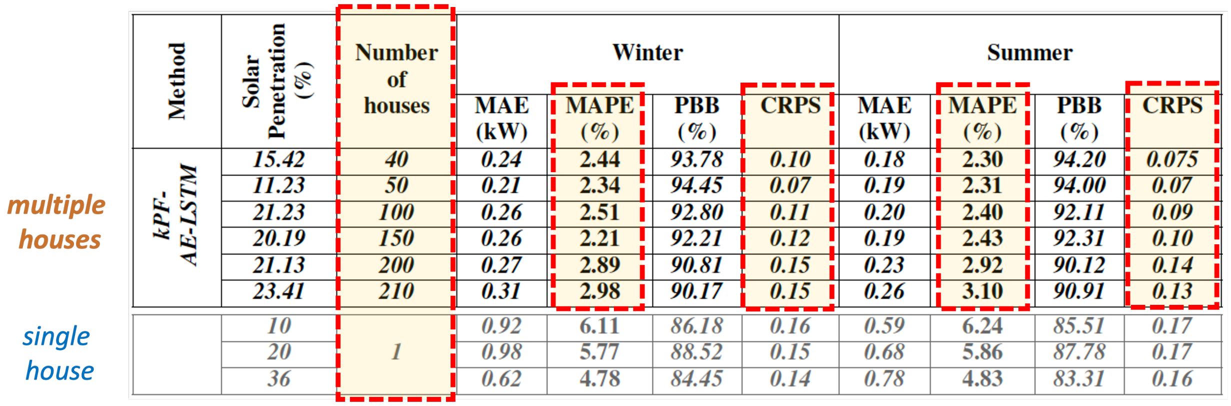
Figure 9: Two forecast models, CLSTM) and PCLSTM, used for benchmarking.

improvement in forecast performance of VRNN over CLSTM and PCLSTM, across all the metrics, with about 50% improvement in MAPE. Moreover, the sequential training approach with VRNN appears to achieve the best-in-class training efficiency (about 4s) while not sacrificing forecast accuracy. Training efficiency gain in our proposed VRNN model (with sequential training) is also associated with 50% less computational memory use when compared to conventional VRNN.



Model	Training Loss	Train time /epoch (s)	MAE (kW)	Norm MAE (%)	MAPE (%)	APE IQR (%)	PBB (%)	CRPS
PCLSTM	CRPS	4.1	9.1	10.7	11.2	7.6	74.57	0.25
PCLSTM	NegLL	4	11.2	13.1	13.3	6.9	73.23	0.39
CLSTM	MAE	4.1	10.8	12.6	13.4	13.3	NA	NA
VRNN	Recon+NegLL+Pred error	7.8	5.6	6.5	6.5	4.1	93.23	0.13
VRNN	Sequential training	4	5.6	6.4	6.6	4.1	93.43	0.12

The above results are generated by performing net-load forecast on single (individual) houses with BTM solar in the NEEA dataset. Next, we tested the performance of the proposed VRNN (kPF-AE-LSTM) model against aggregated house net-load forecasting task. The results are shown below:



	Method	Solar Penetration (%)	Number of houses	Winter				Summer			
				MAE (kW)	MAPE (%)	PBB (%)	CRPS	MAE (kW)	MAPE (%)	PBB (%)	CRPS
multiple houses	kPF-AE-LSTM	15.42	40	0.24	2.44	93.78	0.10	0.18	2.30	94.20	0.075
		11.23	50	0.21	2.34	94.45	0.07	0.19	2.31	94.00	0.07
		21.23	100	0.26	2.51	92.80	0.11	0.20	2.40	92.11	0.09
		20.19	150	0.26	2.21	92.21	0.12	0.19	2.43	92.31	0.10
		21.13	200	0.27	2.89	90.81	0.15	0.23	2.92	90.12	0.14
		23.41	210	0.31	2.98	90.17	0.15	0.26	3.10	90.91	0.13
single house		10	1	0.92	6.11	86.18	0.16	0.59	6.24	85.51	0.17
		20		0.98	5.77	88.52	0.15	0.68	5.86	87.78	0.17
		36		0.62	4.78	84.45	0.14	0.78	4.83	83.31	0.16

The results demonstrate an expected performance improvement with aggregation over multiple houses, compared to the single house scenario. MAPE values drop below 3% across different aggregation levels, with the lowest recorded value of 2.21%, while the CRPS values improve too, achieving within 0.14 across summer scenarios.

Theoretical Analysis: The trained VRNN model is contingent upon the quality and/or availability of the various input data, including, but not limited to various meteorological (e.g., temperature, humidity, etc.) information. Relative importance of meteorological information, especially ambient temperature, in load forecasting has been highlighted in [28, 38], where the authors also point to the data quality, reliability, and availability issues with the meteorological information. In this work, we adopt a nonlinear sensitivity analysis based approach to theoretically estimate the impact of input uncertainties on the forecast output. Conventional sensitivity analysis approaches in neural network, e.g., NeuralSens package [33], attempt to quantify the relative influence of various input data and network parameters on the model output. However, these approaches rely on partial derivatives based first-order sensitivity analysis which fails to capture inherent nonlinear relationships between the input (and network parameters) and the output. Instead, we adopt a data-driven empirical approach to nonlinear sensitivity analysis, which uses the pre-trained forecast model and the available training data to identify the (nonlinear) input-output sensitivities.

Nonlinear Sensitivities: Consider a pre-trained neural network $G_\Theta : X \mapsto Y$, parameterized by the vector Θ, which maps p-dimensional input $X \subseteq \mathbb{R}^p$ to some scalar output (forecast) $Y \subseteq \mathbb{R}$. Consider from the training dataset N pairs of input $x_n \in X$ and output $y_n \in Y$, such that:

$$\forall n = 1, 2, \ldots, N: \quad y_n = G_\Theta(x_n)\,.$$

We are interested in understanding the changes in the model output due to uncertainties and/or corruption in the input data. Considering the perturbed input $\widetilde{x}_n = x_n + \Delta x_n$, where Δx_n is the perturbation vector, we calculate the new output $\widetilde{y}_n = y_n + \Delta y_n$ as:

$$\forall n: \quad \widetilde{y}_n = G_\Theta\left(\widetilde{x}_n\right) \implies \Delta y_n = G_\Theta\left(\widetilde{x}_n\right) - G_\Theta\left(x_n\right) \approx \left.\frac{\partial G_\Theta}{\partial x}\right|_{x_n} \Delta x_n + \Delta x_n^\top \left.\frac{\partial^2 G_\Theta}{\partial x^2}\right|_{x_n} \Delta x_n + H.O.T$$

where we use Taylor series expansion; the terms $\partial G_\Theta / \partial x$ and $\partial^2 G_\Theta / \partial x^2$ refer to the first- and second-order sensitivities, respectively; while *H.O.T* refers to collective higher-order terms. Conventional sensitivity analysis approaches in neural network ignore the second- and higher-order terms, which would fail to capture the relationships that are truly nonlinear (as we illustrate later).

Once these sensitivities are computed empirically (as described below), it gives a way to theoretically compute the bounds on the tolerable uncertainties/noise in the input data.

Theorem 1. *Let there exist a non-negative vector $A \in \mathbb{R}^p$ and a positive semi-definite matrix $B \in \mathbb{R}^{p\times p}$ such that the perturbations in the p-dimensional input vector (denoted by $\Delta x \in \mathbb{R}^p$) and the model output (denoted by $\Delta y \in \mathbb{R}$) satisfy the following relationship:*

$$|\Delta y| = A^\top |\Delta x| + |\Delta x|^\top B |\Delta x|\,. \tag{11}$$

Then for every $\alpha > 0$, the output error will be bounded by α, i.e., $|\Delta y| < \alpha$, if either (or both) of the following sufficient conditions hold on the input perturbation $|\Delta x|$:

1. *(ellipsoidal bounds)* $\left\|\Delta x + \frac{1}{2}B^{-1}A\right\|_2 \leq \sqrt{\left(\alpha + \frac{1}{4}A^\top B^{-1}A\right)/\lambda_{\max}(B)}$
2. *(infinity-norm bounds)* $\|\Delta x\|_\infty \leq \left(\sqrt{\|A\|_1^2 + 4p\alpha\lambda_{\max}(B)} - \|A\|_1\right) / \left(2p\lambda_{\max}(B)\right)$

where $\|\cdot\|_{1/2/\infty}$ refer to the $1/2/\infty$-norms, respectively, and $\lambda_{\max}(\cdot)$ the maximum eigenvalue.

Proof. We only provide a sketch of the proof here. Since B is positive semi-definite, there exists a matrix $M \in \mathbb{C}^{p\times p}$ referred to as the square-root of B, such that $B = M^\top M$. Now, note that the input-output relation in (11) can be expanded as:

$$|\Delta y| = \left\|M\left(\Delta x + B^{-1}A\right)\right\|_2^2 - \frac{1}{4}A^\top B^{-1}A \leq \lambda_{\max}(B)\left\|\Delta x + B^{-1}A\right\|_2^2 - \frac{1}{4}A^\top B^{-1}A$$

which yields the first condition. For the second, we derive the following (Lemma 1 in [24]):

$$|\Delta y| \leq \|A\|_1\|\Delta x\|_\infty + p\lambda_{\max}(B)\|\Delta x\|_\infty^2.$$

Enforcing the inequality $\|A\|_1\|\Delta x\|_\infty + p\lambda_{\max}(B)\|\Delta x\|_\infty^2 \leq \alpha$ yields the second condition. □

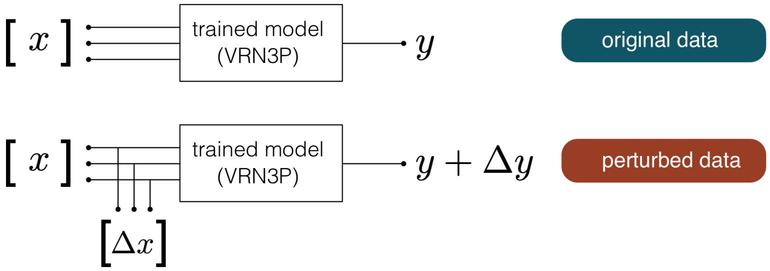


Figure 10: Illustrative description of the experimental procedure to calculate the nonlinear sensitivities.

Algorithmic Steps: The result above provides tolerable bounds on the input perturbation such that the output error is contained within acceptable limits. In order to evaluate the nonlinear sensitivities empirically, we adopt the following experimental procedure (illustrated in Figure 10):

Step 1) (Experimentation) We synthetically inject noise ($\leq 5\%$) to the input training data (e.g., the ambient temperature) and feed it into the pre-trained neural network model VRNN. We record the forecast errors as deviations from the unperturbed output.

Step 2) (Identification) We perform a regression analysis on the input output perturbations data to identify the vector A and the matrix B in the relationship (11).

Step 3) (Validation) We use the learnt (quadratic) functional map to generate forecast error distribution due to noise, and compare that with the actual error distribution. Accuracy of the theoretical estimate is validated by the closeness of the two error distributions (as per Jensen-Shannon distance). In particular, we use the normalized Jensen-Shannon distance (NJSD) [12, 23] as follows:

$$\text{NJSD}(p_1, p_2) = \frac{d(p_1, p_2)}{\sqrt{\ln 2}} \in [0, 1]$$

where p_1 and p_2 refer to the probability distributions of the output perturbations obtained theoretically and via model; and $d(p_1, p_2)$ is the Jensen-Shannon distance. Lower NJSD values denote closer match between the two error distributions, and hence validates the accuracy of the theoretical estimates.

Numerical Validation: For this study, we choose to perturb the ambient temperature values in the training data and plot the resulting perturbation we observe in the model output. To capture the effects on temperature perturbation, outside of the effects of other inputs and situational information, we choose the training data spanned throughout the year and across different operational conditions (weekdays, weekends, winter, summer, etc.). The resulting plot on the left on Figure 11 demonstrates a strong quadratic relationship between the forecast error and the perturbations in the temperature data. Specifically, the relationship remains invariant (unchanged) throughout the

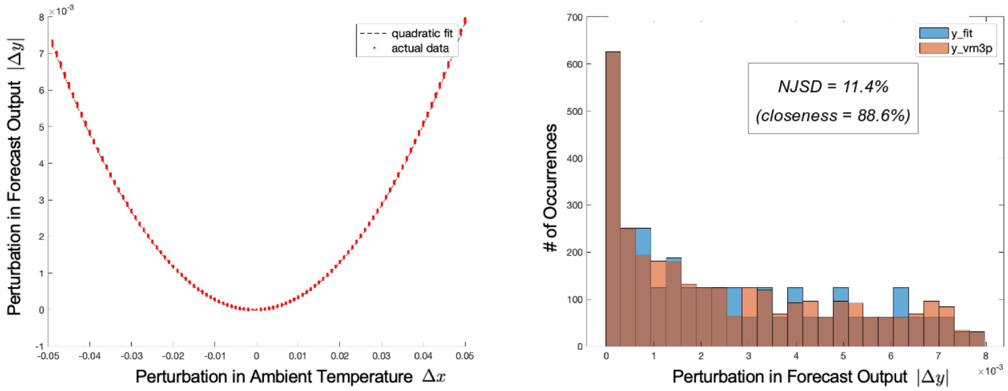


Figure 11: Estimation of forecast output perturbations due to changes in temperature.

year and across various scenarios. This particular finding is in line with the inherently quadratic relationship between net-load and ambient temperature as also reported in previous studies [28, 38]. The input-output data points from Figure 11 (left) are used to run a regression analysis to identify the parameters A and B in (11). In fact, in this case, $A = 0$. The learnt mathematical model is then used to predict the forecast errors due to perturbations in the testing dataset, and compare against the actual forecast perturbations. The resulting forecast error distributions from theoretical estimates and actual model output are compared in the right plot of Figure 11. A NJSD value of 11.4% (i.e., closeness of 88.6%) validates the accuracy of the theoretical predictions.

3.2.5 AI/ML Model: Deployability via Transfer Learning

In order to ensure deployability of the proposed VRN3P framework, we extended the framework by implementing a transfer learning module for online and efficient adaptation of pre-trained networks under unforeseen circumstances, such as high BTM solar penetration. In BP2, the project team implemented transfer learning modules on the VRN3P architecture and performed the validation of transfer learning capability on the test-feeders developed with utility (PGE) data. The following represent the two datasets used for the transfer learning performance validation:

- Source Data: the aggregated net-load profile corresponding to the 20% BTM solar penetration test-case scenario, and associated weather data (temperature and humidity), were used to train the source VRN3P model.
- Target Data: the aggregated net-load profile corresponding to the 30% BTM solar penetration test-case scenario, and associated weather data (temperature and humidity), were used to train the target VRN3P model.

Two distinct scenarios were considered for the transfer learning problem. One is in which full length of target dataset was available for training but only parts of the forecast model is re-trained. The other scenario is in which only a small part of the target data is available for training but the full forecast model is re-trained with warm start from the source model.

Scenario-I: Transfer Learning on Partial Model. In this scenario, only a small fraction of the forecast model is re-trained, while the full length of the target dataset is used for training. Fig. 12 shows the basic concept of the transfer learning exercise adopted in scenario-I. The VRN3P architecture has two main components (or, modules): an inner module of convolutional autoencoder (AE), and

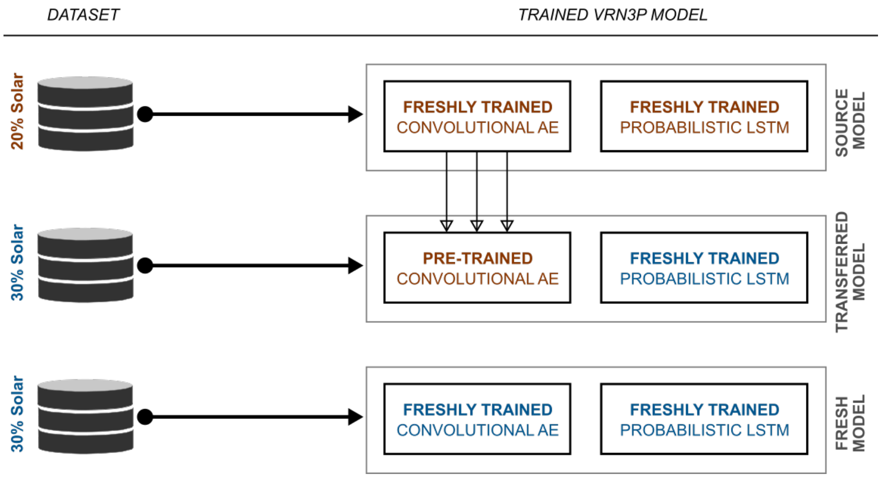
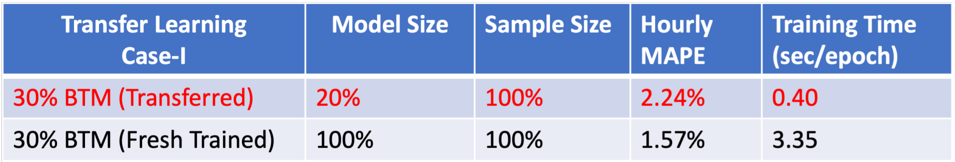

DATASET
TRAINED VRN3P MODEL
20% Solar
FRESHLY TRAINED CONVOLUTIONAL AE
FRESHLY TRAINED PROBABILISTIC LSTM
SOURCE MODEL
30% Solar
PRE-TRAINED CONVOLUTIONAL AE
FRESHLY TRAINED PROBABILISTIC LSTM
TRANSFERRED MODEL
30% Solar
FRESHLY TRAINED CONVOLUTIONAL AE
FRESHLY TRAINED PROBABILISTIC LSTM
FRESH MODEL

Figure 12: Transferable VRN3P scenario-I: transfer learning on partial model.

an outer module of probabilistic LSTM. It has been shown earlier, e.g., in [42], that typically the inner layers of a deep learning architecture is used to learn fundamental (and generic) features of the input-output functional relationships of similar types, while the features specific to the dataset in question are learnt by the outer layer(s). This suggests that only the VAE is transferable between datasets with different solar penetration, while the stochastic LSTM layer is non-transferable, i.e., it needs to be trained on respective solar penetration datasets. This is described in Fig. 12, where three different trainings are performed: 1) a freshly trained VRN3P model on the source data (20% solar), 2) a transferred VRN3P model on the target data (30% solar), and 3) a freshly trained VRN3P model on the target data (30% solar), for comparison. The following highlights the performance comparison of the transfer learned model and the freshly trained model on the target data.



Transfer Learning Case-I	Model Size	Sample Size	Hourly MAPE	Training Time (sec/epoch)
30% BTM (Transferred)	20%	100%	2.24%	0.40
30% BTM (Fresh Trained)	100%	100%	1.57%	3.35

- Source Model: A VRN3P model was trained and tested on the source data (20% BTM solar) to generate day-ahead forecast, achieving on average 1.48% hourly MAPE over a year.
- Target Model: A comparison was made between the forecast and training performance of the two forecast models on the target data: a freshly trained VRN3P model, and a transferred pre-trained VRN3P model (i.e., re-using the VAE module trained on source data). Below are the highlights of the comparison:
  - The transferred pre-trained model achieved comparable, albeit slightly poorer, forecast performance of 2.24% hourly MAPE with the freshly trained model which achieved 1.57% hourly MAPE on the target data.
  - The total number of trainable weights a VRN3P model is 354,000. On the other hand, the total number of trainable weights in the transferred pre-trained VRN3P model is 70,800, while it borrows/transfers the other 283,200 weights from the source model, resulting in 80% reduction in trainable weights.

  - The freshly trained VRN3P model took 3.35 seconds/epoch to be trained on a i5-8 GB CPU, whereas the transfer-VRN3P took 0.56 seconds/epoch to train on the same machine, resulting in 88% reduction in training time, or $8.33\times$ training efficiency.

Scenario-II: Transfer Learning on Reduced Data. In the next scenario, all parts of the forecast model is re-trained, albeit with warm start from the source model, but only a small fraction of the target dataset is used for training. We constructed a lower-volume training dataset which consists of 1 week of data from each month of available training dataset, representing a 75% reduction in available training data size. Fig. 13 shows the basic concept in which both of the convolutional AE and the stochastic LSTM are re-trained. The following highlights the performance comparison of the transfer learned model in the scenario-II.

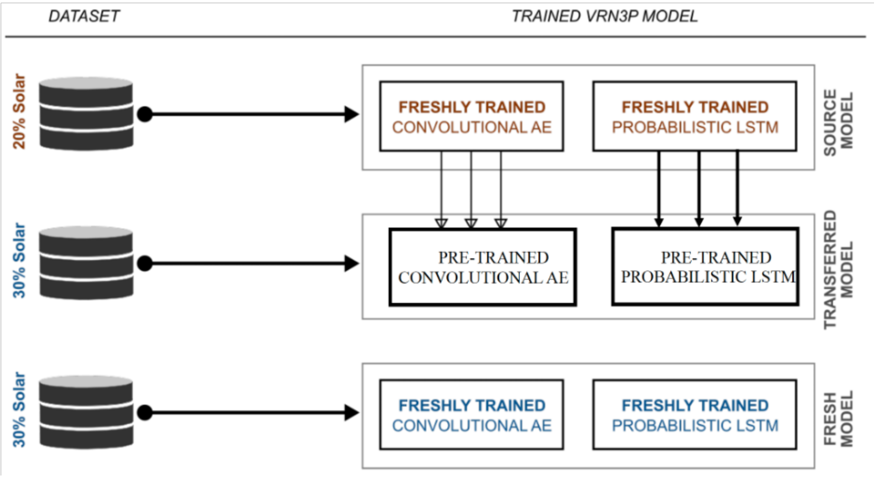

DATASET
TRAINED VRN3P MODEL
20% Solar
30% Solar
30% Solar
FRESHLY TRAINED CONVOLUTIONAL AE
FRESHLY TRAINED PROBABILISTIC LSTM
SOURCE MODEL
PRE-TRAINED CONVOLUTIONAL AE
PRE-TRAINED PROBABILISTIC LSTM
TRANSFERRED MODEL
FRESHLY TRAINED CONVOLUTIONAL AE
FRESHLY TRAINED PROBABILISTIC LSTM
FRESH MODEL

Figure 13: Transferable VRN3P scenario-I: transfer learning on partial model.

Transfer Learning Case-II	Model Size	Sample Size	Hourly MAPE	Training Time (sec/epoch)
30% BTM (Transferred)	100%	25%	2.15%	0.56
30% BTM (Fresh Trained)	100%	100%	1.57%	3.35

- The transferred pre-trained model achieved comparable forecast performance of 2.15% hourly MAPE, compared to 1.57% hourly MAPE achieved by the freshly trained model.
- The total number of trainable weights remained the same in both models (i.e., 354,000), but only 25% of the target dataset were used to train the transfer-VRN3P model.
- The transfer-VRN3P in scenario-II took 0.56 seconds/epoch to train on the same machine (a i5-8 GB CPU), resulting in $6\times$ training efficiency, which was slightly less than the training efficiency achieved in scenario-I.

3.2.6 AI/ML Model: Data Resolution and Model Complexity

While higher complexity in the AI/ML model allows the ability to learn intricate input-output relationships in the training data, thereby improving model output (and forecast accuracy), that comes with the cost of higher data needs to learn effectively and avoid overfitting. Simpler models with fewer learnable parameters, on the other hand, can perform well with less data, though they might miss intricate patterns, highlighting a trade-off between model power, data volume, and potential for bias/overfitting. This trade-off was explored by the project team within the Net-Load Forecasting Prize competition, hosted on the Solar Arbiter Platform. This participation was outside of the main competition, but in a similarly emulated environment to test the performance of the VRN3P forecast models.

Reduced (or Low Resolution) Data: The Net-Load Forecasting Prize made available for training time-series net-load data from four different locations with vaying BTM solar penetration levels: HI (150% solar in terms of annual peak load), GA (63% solar), OR (35%) and TX (19%). The data were sampled at 1-hour and made available for training from a full one year. Compared to the PGE datasets, which were 15-min sampled, the forecasting competition data represented a 25% reduction in data volume available for training.

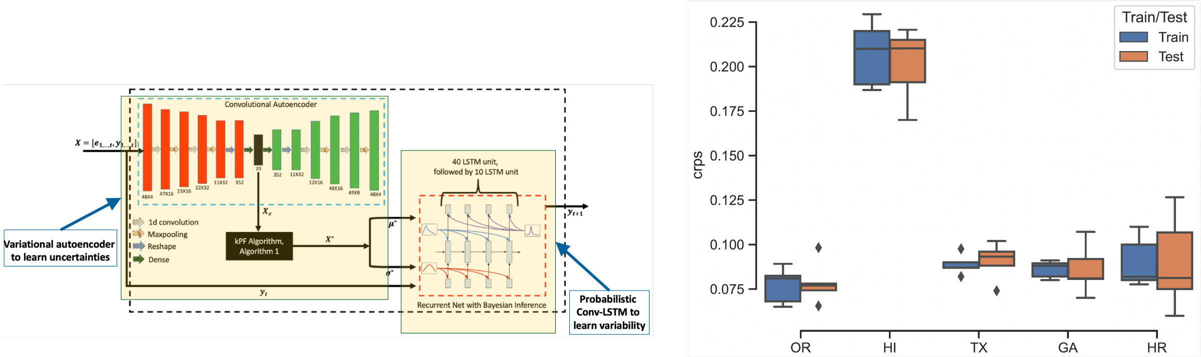


Figure 14: The proposed kPF-AE-LSTM architecture (architecture-1) demonstrated relatively poor CRPS scores on the low-resolution (1-hour) data.

Performance Comparison (Simple vs. Complex Models): Noticing the reduction in available training data volume, the project team considered two possible architectures for the forecasting model: one the proposed kPF-AE-LSTM model (architecture-1), and the other a simpler P-CLSTM model without the autoencoder (architecture-2). Each of these models were then compared against a probabilistic historical reference (HR) forecast model. When testing the kPF-AE-LSTM architecture (architecture-1) on the forecasting competition data, CRPS scores were in the range of 0.075 to 0.1 for three locations (in OR, TX, GA) while in the range of 0.2 in HI, as shown in Figure 14. In comparison, the simpler P-CLSTM model without the AE module (architecture-2) achieved better forecast accuracy, with CRPS scores in the range of 0.05 to 0.075 for three locations in OR, TX, and GA, while in the range of 0.15 to 0.175 in the location in HI, as shown in Figure 15.

This comparison shows the need to keep the forecast models as simple as possible, commensurate with the available data volume for training, to avoid any overfitting issues and thereby degraded forecast performance. Finally, we show snapshots of the performance the model from architecture-2 in the locations in TX (with 19% BTM solar) and GA (with 63% BTM solar), where the team name 'RUBY' was the name of the VRN3P project team. The CRPS scores of the architecture-2 forecast model in HI and OR were, however, high at 0.154 and 0.054.

3.2.7 AI/ML Model: Interactive User Application

VRN3P Application: The VRN3P application was developed on the GridAPPS-D™ platform [31] for distribution system operators. GridAPPS-D™ is a platform sponsored by the U.S. DOE's Office

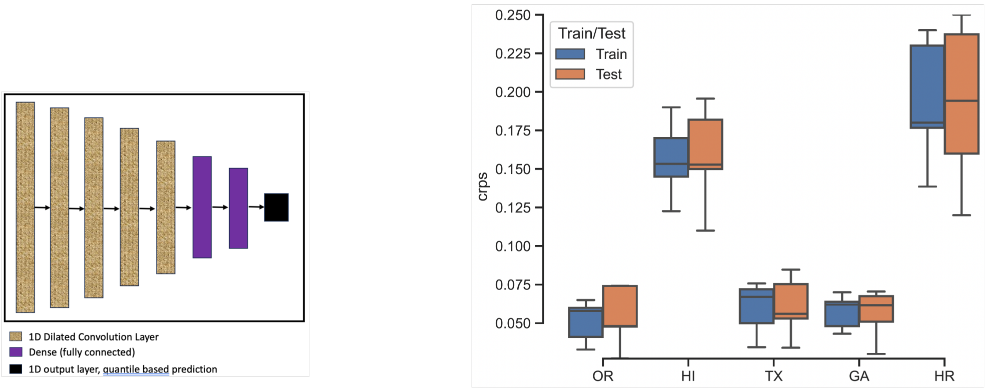
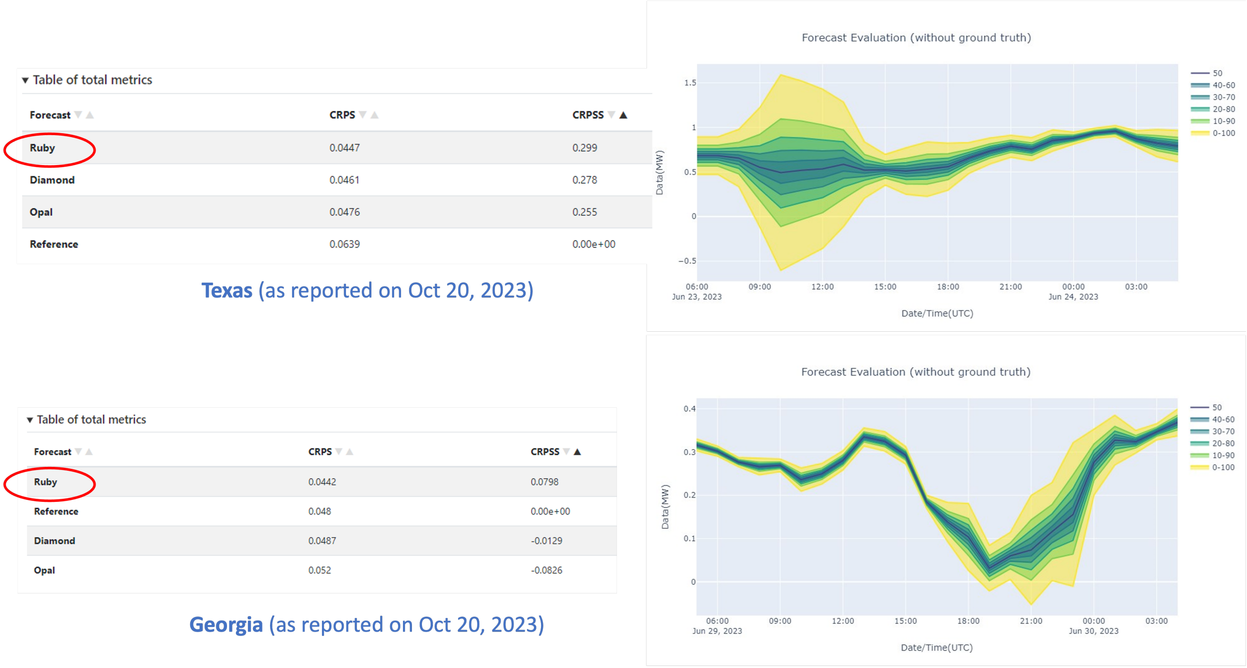


Figure 15: The simpler PCLSTM architecture (architecture-2) demonstrated relatively improved CRPS scores on the low-resolution (1-hour) data.



▾ Table of total metrics

Forecast	CRPS	CRPSS ▲
Ruby	0.0447	0.299
Diamond	0.0461	0.278
Opal	0.0476	0.255
Reference	0.0639	0.00e+00

Texas (as reported on Oct 20, 2023)

▾ Table of total metrics

Forecast	CRPS	CRPSS ▲
Ruby	0.0442	0.0798
Reference	0.048	0.00e+00
Diamond	0.0487	-0.0129
Opal	0.052	-0.0826

Georgia (as reported on Oct 20, 2023)

Figure 16: Forecast performance of the architecture-2 in TX and GA, as reported on a day in Oct, 2023.

of Electricity, Advanced Grid Research program. The open-source platform supports accelerated development and deployment of portable applications for advanced distribution management systems. It is distributed using Docker containers. GridAPPS-D™ incorporates standardization of data models, programming interfaces, and the data exchange interfaces. The Common Information Model (CIM), as defined in the IEC 61970 and IEC 61968 series of standards, can be used to describe all power system models and enables the exchange of models across compliant applications and services. Using the set of standardized model queries provided by the PowerGrid Models API, a GridAPPS-D application is able to scale seamlessly across different network models with no modifications to the application code. The GridAPPS-D Platform and GridAPPS-D APIs provide a standardized method for interfacing with power system modelling data, real-time simulation data, historical data, and logging information. Each of these APIs abstracts the specifics of the database, and enable simple queries through a set of standardized messages formatted as JSON strings.

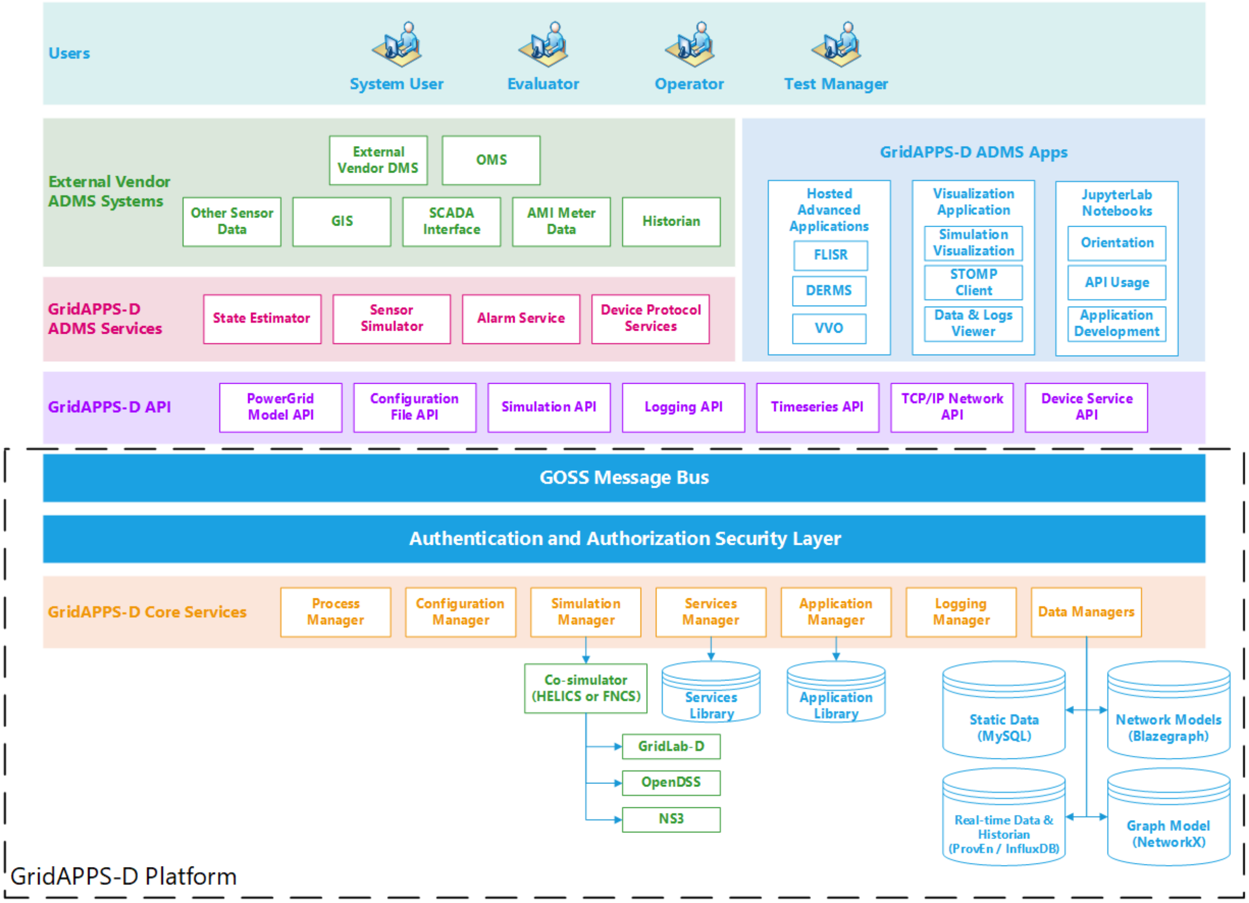


Figure 17: The architecture of the GridAPPS-D™ development ecosystem.

Figure 17 shows the overall development ecosystem of GridAPPS-D™ which allows rapid integration of advanced applications and services through several APIs, including:

- *Powergrid Models API* – Allows apps and services to access the power system model data
- *Configuration File API* – Allows apps to set equipment statuses and system conditions
- *Simulation API* – Allows apps to start a real-time simulation and issue equipment commands

- *Timeseries API* – Allows apps to pull real-time and historical data
- *Logging API* – Allows apps to access logs and publish log messages

Of particular interest to VRN3P is the *Timeseries API* which are used to query the time series data store that keeps the measurement data from simulations. The API calls can be used to obtain weather data, measurements from simulation data using measurement mRIDs, equipments commands and other simulation input data, and simulated field data from the Sensor Service.

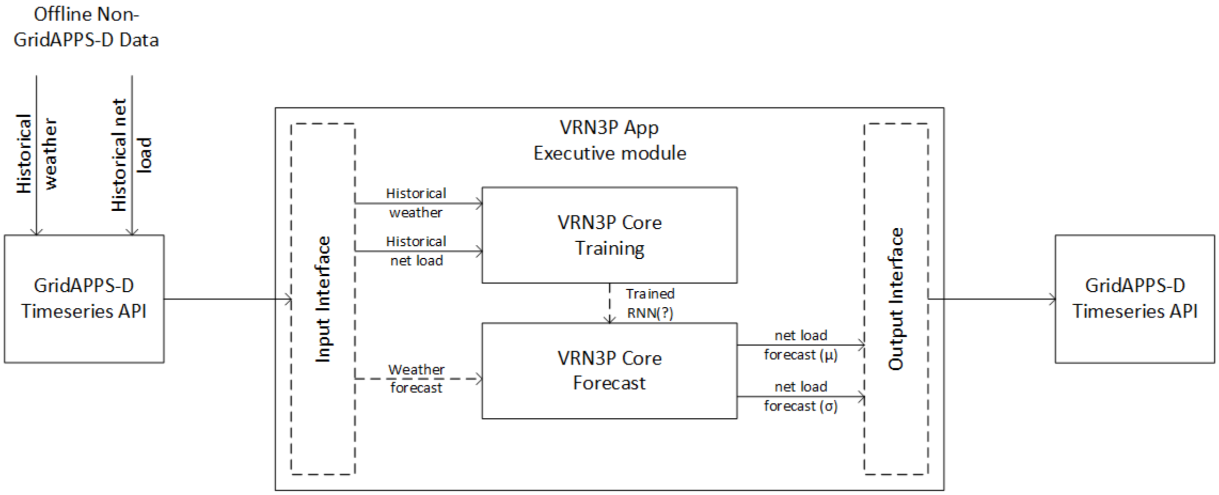


Figure 18: Sketch architecture of the VRN3P GridAPPS-D™ application under development.

Figure 18 shows a sketch of the architecture of the GridAPPS-D application. It has an input and an output interface that use the GridAPPS-D Timeseries APIs to read historical net-load and weather data, as well as publish the probabilistic net-load forecast values (in mean and variance). An application suite has been developed based on the GridAPPS-D™platform. It contains four different Python scripts, as described below. These applications are responsible for reading and storing input data to the GridAPPS-D™Timeseries database, training a deep variational recurrent neural network, performing net load forecasting based on the trained model, and visualizing the results. A short description for each of these programs is provided next.

- `load_ts_ordered.py`: This script is responsible for reading an input CSV file and storing its contents to the GridAPPS-D™Timeseries database, while maintaining the data order of the CSV file. The user is expected to provide the input CSV file in the same folder from which load_ts_ordered.py is called.
- `VRNN_ts_ordered_train.py`: This script contains the necessary code for training the deep variational recurrent neural network that has already been developed as part of the VRN3P project. It receives data from the GridAPPS-D™ Timeseries database through a query. It returns a saved trained ML model under the “trained_model” folder.
- `VRNN_ts_ordered_predict.py`: This script performs the main forecasting task. It receives as input a saved trained ML model under the “trained_model” folder as well as appropriate input information in the same format as the information of the input CSV file. It returns a distribution system net load forecast that is stored on the GridAPPS-D™ Timeseries database and (optionally) in a CSV file.

- `plot_ts_predicted.py`: This script is responsible for visualization. It receives a distribution system net load forecast that is stored on the GridAPPS-D™ Timeseries database as input. It plots (or saves to figure if so desired) the predicted forecast.

Web-Based User-Interface: A barrier to adoption of AI/ML models in grid operational planning stems from a perceived lack of trust in the model, which can be improved if the model outputs are accessible to be explored by domain experts, including scientists and grid operators [2, 4]. The model's performance may fluctuate due to seasonal variations in the input variables, and stakeholders must also assess its reliability in the face of noisy inputs mirroring real-life scenarios. Hence, the process is complex and time-consuming, prompting the need for an approach capable of performing these tasks and enhancing trust in the model's performance. Prior research has shown that visual analytics can significantly enhance trust in model outputs during complex sense-making tasks [9] and is expected to play a critical role in enabling trust-augmented AI/ML applications in energy sector [19].

In this project, we developed a visual analytics-based application `Forte` [2, 3] that empowers users to gain an in-depth understanding of the model's performance, effectively leveraging data visualization techniques to aid informed decision-making in the realm of energy planning and grid operations. First, it enables researchers and scientists in the energy domain to assess net-load variability concerning input variables by comparing model forecasts with actual net-load values across different time periods and seasons. They can gain insights into their impact on model performance by analyzing the effects of variables like temperature, humidity, and apparent power on net load forecasts. Second, `Forte` helps evaluate forecast errors with noisy inputs at different noise levels, thus providing information for improving the model's reliability and robustness in real-world scenarios. This visual analytics-based approach can empower scientists and grid operators to make data-driven decisions, enhancing trust and confidence in the net load forecasting model. It combines interactive visualization with performance metrics to instill greater trust in model outcomes, providing users with the flexibility to probe net load predictions as a function of input variables like temperature and humidity.

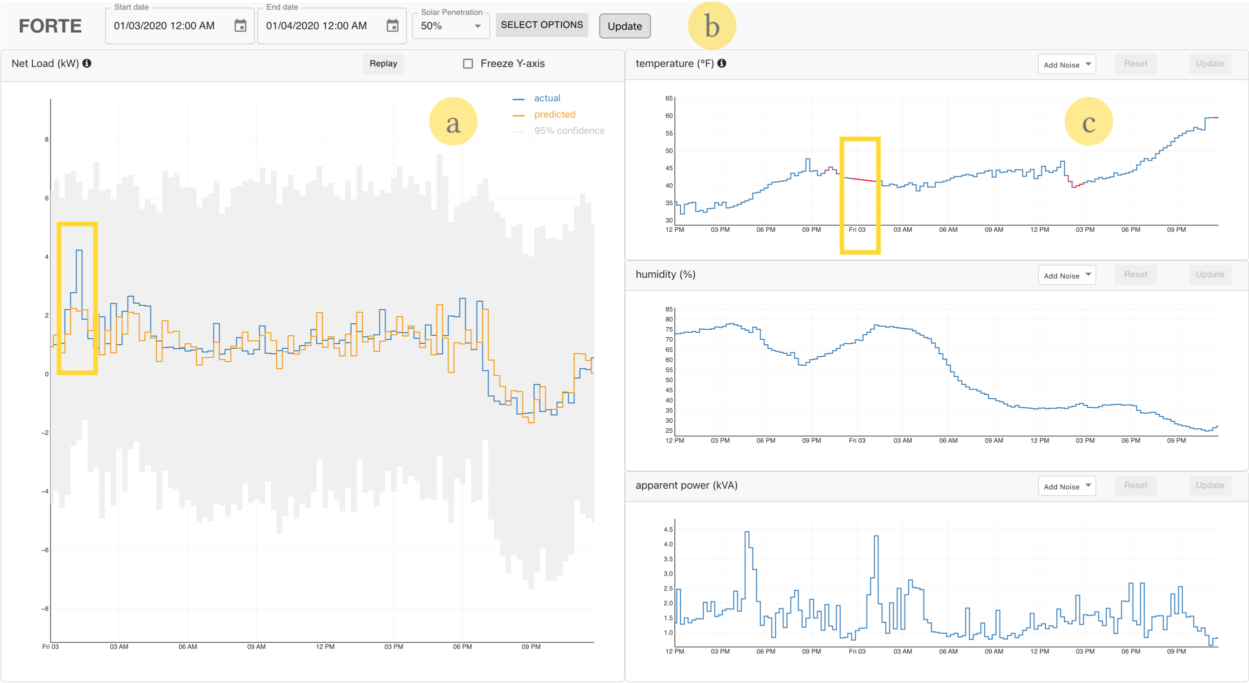
Primarily developed using React.js and D3.js for the frontend, and Flask framework in Python for the backend, `Forte` aims to achieve the following visual analytic tasks (see details in [2, 4]):

- ***T1**: Understand actual and predicted net-load across time periods and BTM solar levels*
- ***T2**: Explore the impact of input variables on net load prediction*
- ***T3**: Augment missing data with background knowledge*
- ***T4**: Design experiments simulating different noisy input scenarios*
- ***T5**: Assess model efficacy across different months and varying levels of noise*
- ***T6**: Compare models across different BTM solar levels and data resolutions*
- ***T7**: Identify hidden temporal patterns in forecast model performance*

Short demonstration videos are available at: link-1 and link-2. Next, we explain our application's design by outlining a few of its high-level goals as examples.

[Goal-1] Understand Net Load Forecasts w.r.t Input Variables: Understanding the inter-dependence between net-load forecasts and input variables is essential for making informed decisions in energy planning and operations. Towards this end, `Forte` integrates three essential components:

3.2 Detailed Results and Discussions



Start date 01/03/2020 12:00 AM
End date 01/04/2020 12:00 AM
Solar Penetration 50%

Figure 19: **Net-load forecasting visual analytic tool (`Forte`):** (a) Our application facilitates the comparison of actual and predicted net load within the selected time frame and solar penetration levels as defined (b) through the Options Selection Area. Further, (c) the highlighted region within the Inputs View Area shows instances of missing temperature data and resultant disagreement between predicted and actual net-load.

[1a] Options Selection Area: As mentioned earlier, net load forecasts can fluctuate based on time periods and solar penetration levels. Accordingly, `Forte` offers these selections prominently at the top, within the Options Selection Area (Figure 19b). Additionally, users can choose solar penetration levels from 0%, 20%, 30%, and 50%. This area provides options for choosing different prediction horizons (15 minutes or 24 hours ahead) and input variables (temperature, humidity, apparent power, etc.) tailored to user preferences.

[1b] Net-Load View Area: This component facilitates a direct comparison between the actual net load and the predicted net load for the chosen time period and solar penetration level, as selected within the Options Selection Area **(T1)**. This visual representation employs a blue line to depict the actual net load and an orange line to depict the predicted net load (Figure 19a). By hovering on the icon button atop this area, forecast accuracy measures such as Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) are revealed. Since our net load forecasting model produces a probabilistic forecast, we additionally present the 95% confidence interval for this forecast, indicated by a subtle, shaded grey area. When users modify the options, we noted that the Y-axis within this area might shift due to value changes, impeding the observation of variations across distinct time periods or solar penetration levels. This problem can be alleviated using the "Freeze Y-axis" option, which, as the name suggests, freezes the Y-axis at the current values and plots the new values based on the frozen axis. Additionally, changes can be tracked using the Replay button, which showcases net load changes through a slower animation (≈ 10s).

[1c] Inputs View Area: Located on the right-hand side of the application, the Inputs View Area displays the selected inputs and their respective values during the chosen time period (Figure 19c). It

also shows some of the historical data used while generating the forecast for this period. This visualization aids in establishing correlations between weather data and the agreement/disagreement observed between the actual and predicted net load, thereby impacting model performance **(T2)**. Nonetheless, weather data might feature gaps for specific time spans, which are addressed through linear interpolation connecting the nearest available data points. These interpolated points are indicated in red, and users have the flexibility to drag and adjust them based on their expertise **(T3)**. The data quality, denoting the percentage of missing data, can be accessed by hovering over the icon button atop each input variable. However, if the users do not trust the quality of the available weather data, they can apply a uniform noise of 5% or 10% via the "Add Noise" option for each input variable. All these changes are reflected on the Net Load View Area once users hit the "Update" button. Thus, `Forte` begins with simple visualization and default settings, easing the learning curve as users delve into advanced features.

[Goal-2] Compare model performance w.r.t Noisy Inputs: The AI/ML forecasting model's responses varied with the noise levels in the input variables. A separate linked page was developed with the following components to allow users the opportunity to experiment with diverse noise scenarios.

[2a] Experiment Design Area: Users can generate simulated noisy inputs for varying dates spanning multiple months and a specific input variable. This area empowers users to select their preferred input variable (temperature, humidity, apparent power), set start and end dates, and designate desired months for introducing noise **(T4)**. The area also offers the flexibility to add or subtract a uniform noise (ranging from 1% to 30%) from the original inputs or use a combination thereof (Figure 20a). The users can add a name and short description of the experimented scenario for future reference, and our application will show an estimated time for completing this experiment.

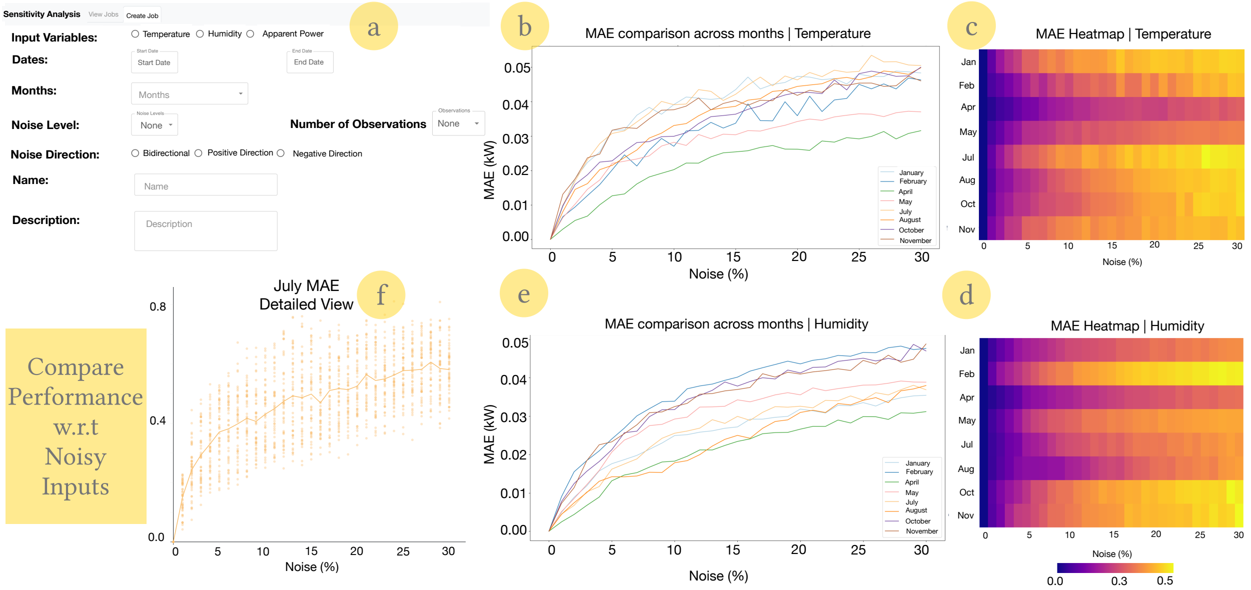


Figure 20: `Forte` **Experiments:** (a) `Forte` enables the design of experiments through the creation of noisy inputs using various factors; and the results (error rates) can be cross-compared across various months for both the input variables of (b, c) temperature and (d, e) humidity; (f) with the option to view detailed observations for each month. These insights could help the user (a grid operational planner) to better prepare ahead of any impending weather events (e.g., heat/cold wave).

[2b] Experiments View Area: Users can select any of the completed experiments/jobs from the left-hand side navigation bar. Each experiment initially displays two line charts depicting the error metrics MAE and MAPE (Figure 20b, 20e). These charts comprise lines corresponding to the months chosen during the experiment design. Each line illustrates the deviation in error metrics from their baseline values (established at 0% noise or no noise) **(T5)**. We offer an alternative visualization in the form of a heatmap, illustrating the deviation in MAE from the baseline values for each month. Based on initial feedback, users found this heatmap particularly useful for comparing the model's sensitivity across different months (Figure 20c, 20d). In addition to this, users may want to explore the error rates for each month. Hence, we also include two scatterplots for each month (for each of the error metrics), which show the error rates for each of the observations (Figure 20f). This scatterplot is then augmented with a line showing the average error rate for that month across different noise levels, mimicking the corresponding line in the first line chart.

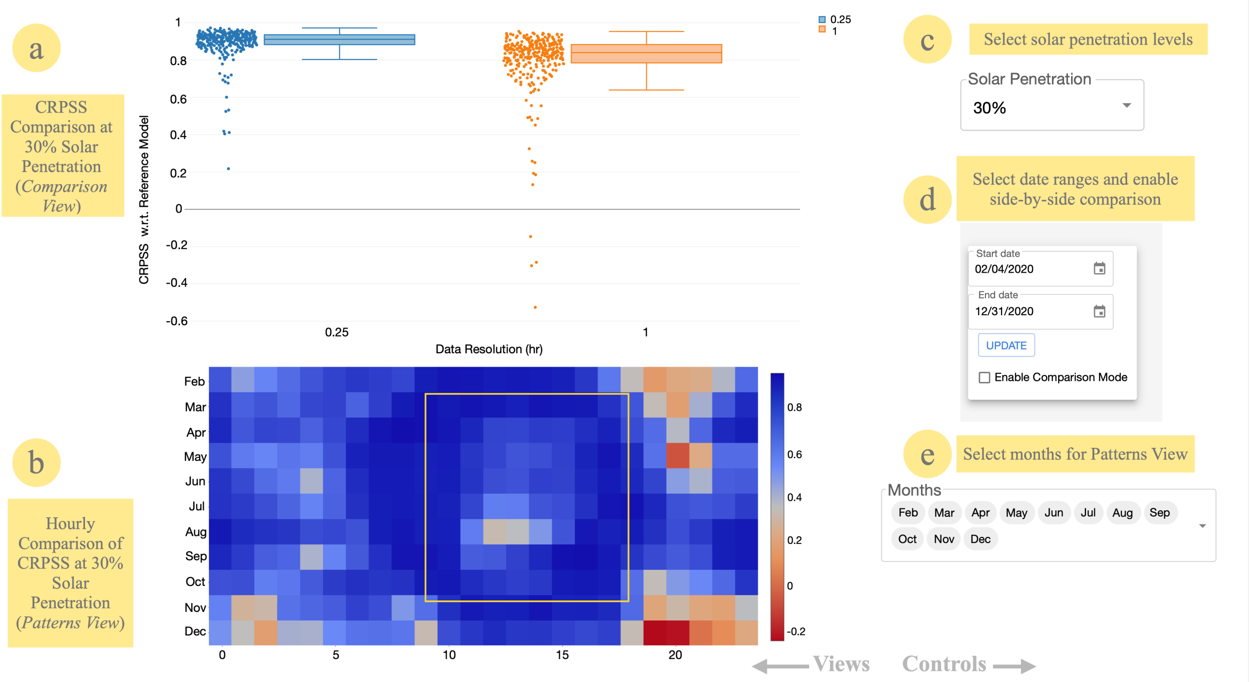
Building Trust via Input-Output Sensitivity: The experiments view area helps to understand the model's performance when faced with noisy inputs and input perturbations, and, in the process, improves users' trust in the model [2]. For example, as shown in Figure 20c, the model displayed heightened sensitivity to noisy data during January and July, across numerous noise levels—albeit with exceptions. In contrast, the model exhibited the least sensitivity during April and May (Figure 20b and 20c). The observed variations in the model's sensitivity to noisy perturbations in temperature data across different months, can be attributed to the influence of seasonal weather variations on usage of electricity **(T5)**. For example, typically the heating and cooling load – which drives the residential energy demand – typically peaks during the coldest (e.g., January) and the hottest (e.g., July) months, thereby ensuring heightened sensitivity of net load to temperature variations. In contrast, sensitivity of residential energy usage to temperature perturbations remain low in shoulder months (e.g., April and May) with typically milder weather.

[Goal-3] Uncover Hidden Patterns in Model Performance: By comparing performance of different models across multiple timeframes, power scientists can assess the net-load forecasting models' robustness and consistency in capturing both short-term fluctuations and long-term trends, and identify patterns in a model's performance across different months, various hours of the day, and different time periods. For this purpose, a probabilistic convolutional LSTM (architecture-2 in Figure 15) was used, and compared against a reference forecast model that simply utilizes historical input data from the past 30 days to generate probabilistic forecasts for a specific time point. Continuous Ranked Probability Skill Score (CRPSS) was used to evaluate and compare the relative performance of any model (e.g., P-CLSTM) against the reference model [10] across multiple dates throughout the year. A positive CRPSS indicates that the forecast outperforms the reference forecast, whereas a negative value suggests inferior performance. Our application implements multiple coordinated views and components in order to fulfill these tasks:

[3a] Comparison View: In `Forte`, the Comparison View utilizes modified box plots to compare models. Figure 21a depicts CRPSS values on the y-axis and different data resolutions on the x-axis. Positive CRPSS indicates superior performance of the model under study (in this case, the PCLSTM) against the reference model. Users can utilize the solar penetration level filter to compare performance across various levels (20%, 30%, 50%) (Figure 21c) **(T6)**.

[3b] Patterns View: We developed a Patterns View where the users can identify the timeframes where the model underperforms compared to the reference. In this view, `Forte` utilizes a heatmap

3.2 Detailed Results and Discussions



Solar Penetration: 30%

Start date: 02/04/2020

End date: 12/31/2020

□ Enable Comparison Mode

Months: Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

Figure 21: **Results from a case study:** (a), (b), (c) display CRPSS values at varying solar penetration levels, highlighting the model's superior performance with higher-resolution datasets. (d) Additionally, our application reveals insights such as the model's ability to learn and predict diurnal patterns, as evidenced by highlighted box-like patterns.

to depict performance patterns for each month across different hours of the day **(T7)**. The x-axis represents the 24 hours of the day (0-23), while the y-axis displays the months of the year (Feb - Dec) (Figure 21b). Each box in the heatmap denotes the average CRPSS value for each month at each hour, indicated by the color. Darker blues signify more positive CRPSS values, indicating the superior performance of our model compared to the reference at that time. Conversely, darker reds indicate more negative CRPSS values, signifying poorer performance of our model compared to the reference at that time. Users can filter specific months to focus on particular time periods and analyze performance accordingly (Figure 21e).

[3c] Sidebar: The Sidebar facilitates user selection of start and end dates to filter results across all views, enabling focus on specific date ranges **(T7)** for comparing model performance within those periods (Figure 21d). Additionally, `Forte` offers a comparison mode toggle in the Sidebar. Enabling this mode updates both views to display box plots and heatmaps for all solar penetration levels side by side, aiding in point-to-point comparison and identification of performance patterns across different solar penetration levels.

Building Trust via Revealing Temporal Patterns: Upon closer inspection, the CRPSS value distribution in the heatmap at a 20% solar penetration level (Figure 21b) within the Patterns View in `Forte` reveals useful temporal patterns to understand model's performance. Most heatmap boxes displayed varying shades of blue, indicating superior model performance compared to the reference across most months and hours. On closer look, the user may note a box-like pattern in the heatmap, revealing enhanced performance during morning hours (8 am to 10 am) from April to

September, followed by a decline during midday (11 am to 4 pm), and another spike in the evening (4 pm to 6 pm) during these months. This pattern suggested that the model effectively captured diurnal variations in net load data and adjusted predictions accordingly **(T7)**.

3.2.8 Techno-economic Benefits

To quantify the financial impact of day ahead net-load forecast errors, we consider as the benchmark the day ahead scheduling reserves calculated for use in a DOE Water Power Technology Office (WPTO) study aimed at identifying the drivers of locational marginal price (LMPs) within the 2020 CAISO market, called the WPTO Value Drivers study [25].

- The LMPs were calculated using a production cost model (PCM). The PCM takes as input the 8760 hourly time series of wind and solar power production at the plant level, and net load (native load minus BTM solar) at the balancing authority (BA) level. The PCM reserves capacity for up and down regulation and load following to absorb intra-hour forecast errors. These reserves impact LMPs.
- The PCM was developed for WPTO. The generator and load inventory is based on the WECC 2030 Anchor Data Set. Power plants were back-cast to reflect inventory in 2020. Data for the PCM was pulled from EIA930 (hourly BA generation and net load), and fuel prices from EIA (Henry Hub). Emission allowance prices were pulled from CAISO Oasis for 2020.
- The result of this study is a well-calibrated PCM which incorporates reserve requirements at the BA level and is validated with 2020 CAISO LMPs.

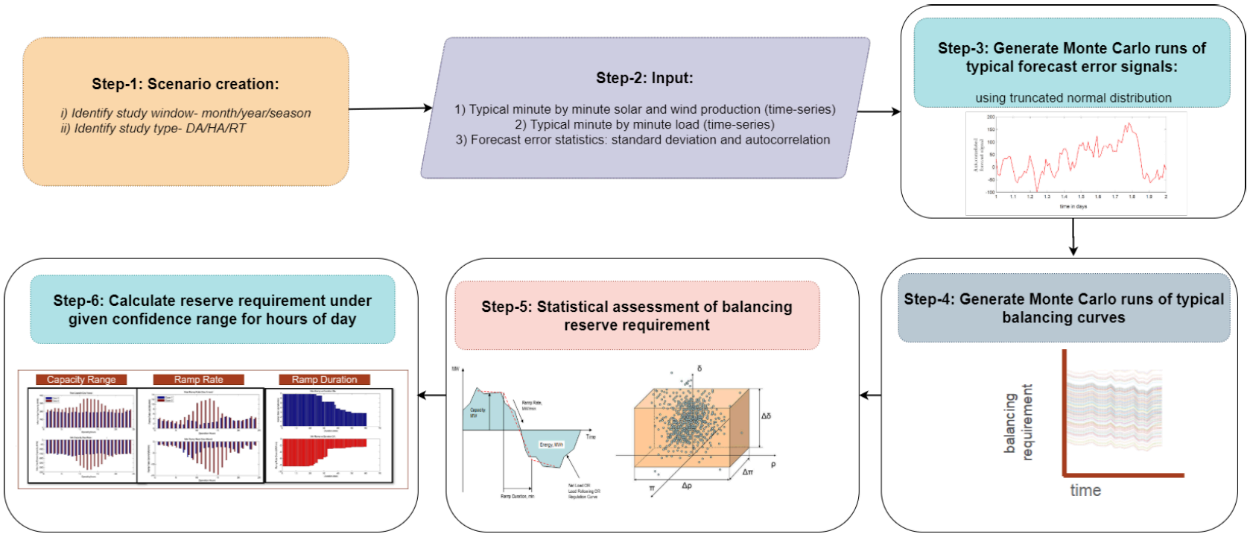

Step-1: Scenario creation:
i) Identify study window- month/year/season
ii) Identify study type- DA/HA/RT
Step-2: Input:
1) Typical minute by minute solar and wind production (time-series)
2) Typical minute by minute load (time-series)
3) Forecast error statistics: standard deviation and autocorrelation
Step-3: Generate Monte Carlo runs of typical forecast error signals:
using truncated normal distribution
time in days
Step-4: Generate Monte Carlo runs of typical balancing curves
balancing requirement
time
Step-5: Statistical assessment of balancing reserve requirement
Step-6: Calculate reserve requirement under given confidence range for hours of day
Capacity Range
Ramp Rate
Ramp Duration

Figure 22: PNNL GRAF-Plan tool estimates the reserve requirements for each Balancing Authority [13].

Within the WPTO Value Drivers study, the reserves for each balancing authority (BA) are calculated in the PNNL tool GRAF-Plan (Figure 22), and are based on observed forecast errors for BA-level net-load (native load minus BTM solar), utility scale solar, and utility scale wind [13]. Please refer to [29] for a detailed discussion of the reserve calculation methodology. The reserves are calculated by re-creating synthetic forecasts of net-load, wind, and solar by Monte Carlo sampling of empirically derived forecast error distributions for each resource. The aggregation of errors between look-ahead time intervals (e.g., day-ahead to real-time) provides the scheduling reserve imbalance signal for which capacity must be reserved. The GRAF-Plan tool compresses the signals across 30 sample draws for each month to provide the 95% capacity requirement throughout the 24 hours in a day for that month (same 24 values for one month). GRAF-Plan creates probabilistic forecasts by drawing samples from a truncated normal distribution and autocorrelating the errors prior to adding the errors to the bulk net-load time series. In contrast to the forecast error reconstruction of GRAF-Plan, the VRN3P day-ahead net-load forecast model produces N draws from the probabilistic posterior distribution of the forecast. Incorporation of the VRN3P forecast results into GRAF-Plan is done by producing probabilistic forecasts in this model and importing the N sample draws into the GRAF-Plan tool.

Reconstruction of the net-load forecasts within GRAF-Plan requires:

- Mean of normalized time series
- Standard deviation of normalized time series (standard deviation is divided by the max net-load in a year)
- Autocorrelation of normalized time series for lag of one (this accounts for “red noise” in errors)

In the WPTO Value Drivers study, the 5-min-ahead, 1-hour-ahead, and 24-hour-ahead forecast error statistics were drawn from observed forecast errors within the CAISO territory, and applied to all BAs in the WECC region.

In this analysis, the financial impact of improved day-ahead net-load forecast is defined as the percent reduction in day-ahead scheduling reserves.

The experiment to benchmark the financial impact of the VRN3P model was conducted as follows:

1. BA-level day-ahead scheduling reserves: Compare the “state-of-the-art” BA-level forecast (2020 CAISO forecast net-load forecast statistics) to VRN3P within a full service territory (quantitative comparison)
   - The benchmark 2020 reserves are calculated using the 2020 CAISO net-load forecast error statistics and 2020 EIA930 hourly net-load for PGE territory.
   - The VRN3P 2020 reserves are calculated by generating net-load probabilistic forecasts using VRN3P on 2020 EIA930 hourly net-load for PGE territory.
2. Feeder-level day-ahead scheduling reserves: Compare the VRN3P model on progressively higher BTM solar penetration test scenarios (qualitative comparison): 20%, 30%, and 50% BTM solar scenarios.

BA-level Day-Ahead Scheduling Reserves Figure 23 shows one day in April. This illustrates the probabilistic forecast of VRN3P, showing that it closely follows the actual net-load time series. In Figure 24, we see that the hourly day-ahead scheduling reserves in a given month are substantially reduced with respect to the state-of-the-art in day-ahead net-load forecasting at the BA level (from CAISO), when applied to the PGE service territory. In Figure 25, we see that the hourly reserves in a given month are reduced by roughly 60-80% when the VRN3P forecast is used to calculate the necessary Day-Ahead Scheduling Reserves. Tables 4 and 5 show the hourly and monthly average percent improvement of the VRN3P net-load forecast algorithm with respect to the CAISO forecast error statistics when applied to the PGE area load from EIA930 in 2020. On average, the percent improvement for both scheduling up and down reserves is above 80%.

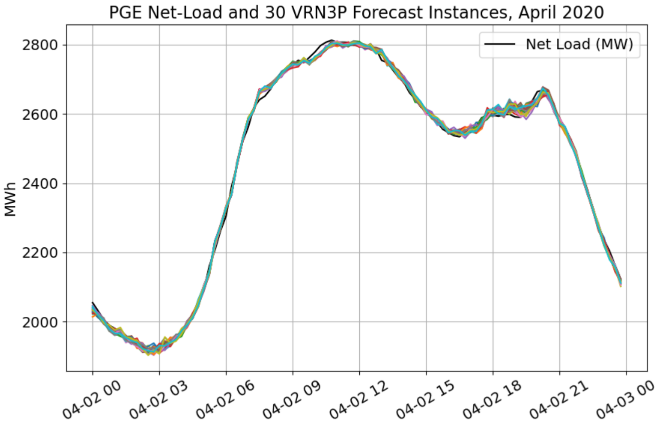


Figure 23: One day in April

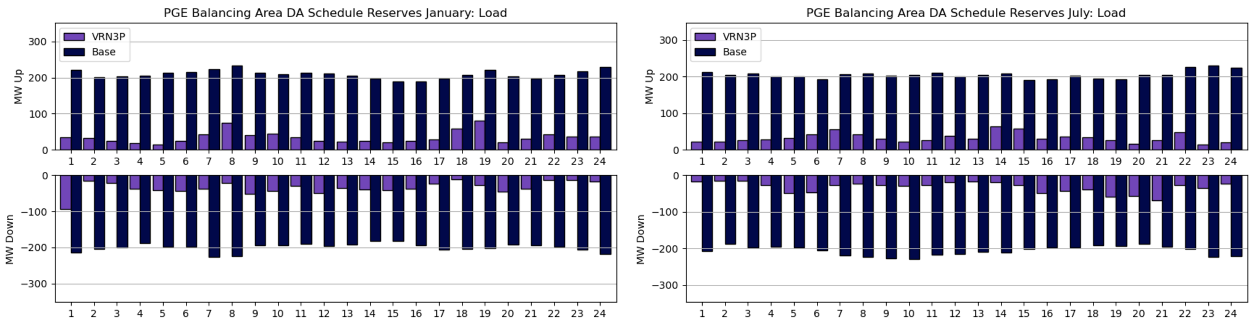


Figure 24: Day-ahead Scheduling Reserves in PGE BA. January (left), July (right)

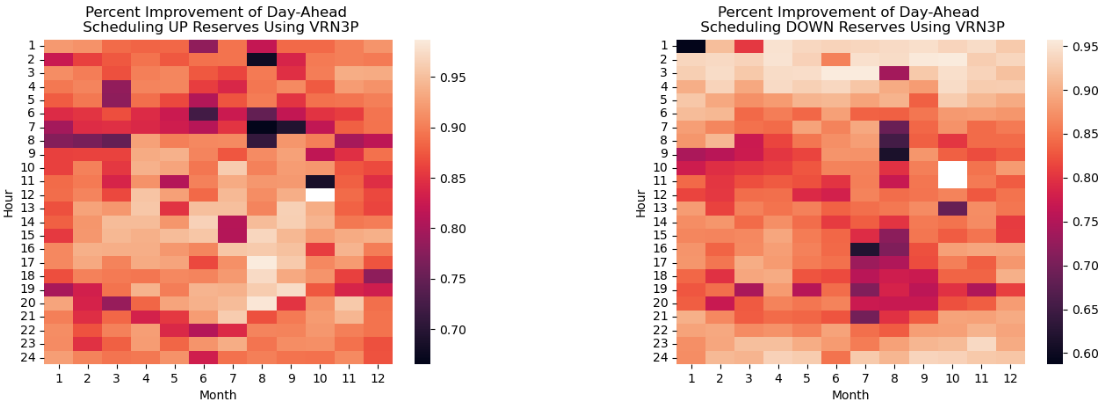


Figure 25: Relative Improvement in Day-Ahead Up (left) and Down (right) Reserves, VRN3P vs CAISO

Table 4: Hourly average percent improvement of VRN3P over CAISO day-ahead net-load forecast

Hour	Up	Down
1	88%	90%
2	86%	93%
3	89%	92%
4	88%	92%
5	87%	90%
6	83%	87%
7	81%	85%
8	84%	83%
9	89%	80%
10	90%	83%
11	87%	83%
12	91%	83%
13	91%	84%
14	92%	84%
15	92%	84%
16	93%	83%
17	92%	84%
18	90%	83%
19	89%	79%
20	90%	82%
21	89%	84%
22	88%	88%
23	90%	90%
24	89%	91%
Mean	89%	86%

Table 5: Monthly average percent improvement of VRN3P over CAISO day-ahead net-load forecast

Month	Up	Down
January	87%	86%
February	88%	86%
March	87%	85%
April	91%	88%
May	89%	86%
June	88%	86%
July	90%	84%
August	88%	80%
September	90%	86%
October	88%	87%
November	89%	87%
December	88%	88%
Mean	89%	86%

Feeder-level Day-Ahead Scheduling Reserves

Figures 26 and 27 show the day-ahead scheduling reserve calculated for the 20%, 30%, and 50% BTM solar penetration scenarios for January and July at one feeder in PGE. The day-ahead scheduling reserves at all penetrations of BTM solar show a diurnal pattern in the reserve estimation imposed by the BTM solar. The magnitude of reserves is larger in the summer months (Figure 27, yet the diurnal duck-curve pattern is strong throughout the year. The normalized 24-hour-ahead net-load forecast error statistics at the BA level are shown in Table 6 alongside the error statistics of the VRN3P algorithm applied to the PGE territory. The VRN3P forecast has resulted in reduced day-ahead scheduling reserves at the BA level for PGE, which is a direct result of the reduction in forecast error.

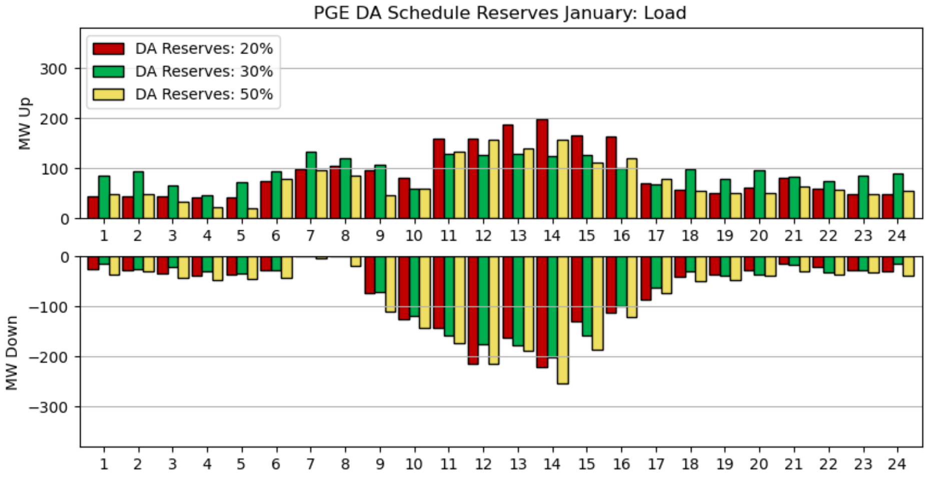


Figure 26: Day ahead scheduling reserves at 20%, 30%, and 50% BTM solar in January for PGE.

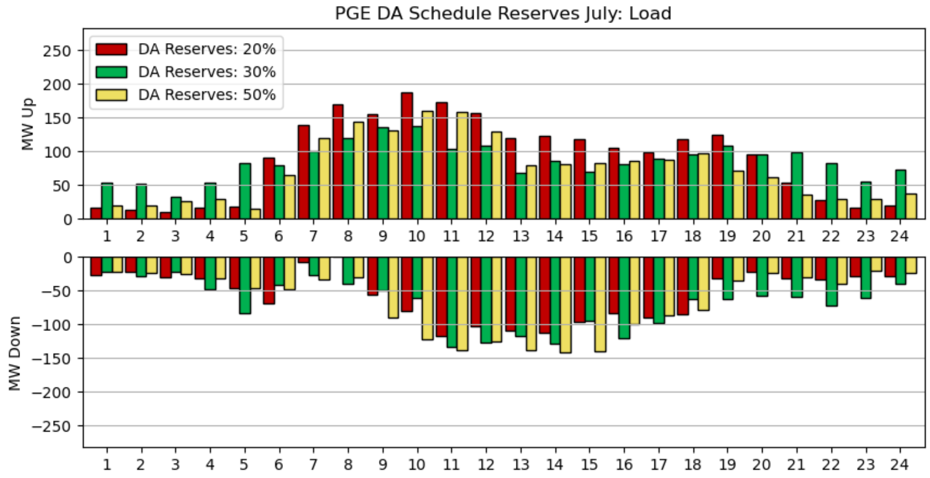


Figure 27: Day ahead scheduling reserves at 20%, 30%, and 50% penetration of BTM solar in July for PGE.

Table 6: Day ahead forecast statistics at the BA level for PGE

	2020 WPTO Study (CAISO)	VRN3P
mean	0	0
standard deviation	2.2%	0.46%
autocorrelation	0.92	0.68

Additional Remarks

The project team faced challenges in quantifying the impact of the improved net-load forecast on the operational savings. Since, as of today, most utilities (including our partner utility PGE) do not use day-ahead feeder-level net-load forecast in operational decisions, it is non-trivial to quantify any improvement in forecasting those values. As such, the project team decided to apply the VRN3P algorithm in forecasting CAISO net-load forecasts at the balancing area level. The results from this study, as reported here, show significant savings in reduced reserves allocation when using VRN3P forecasts, over the CAISO benchmark forecasts. Closer review of these results with stakeholders, however, is needed to understand its true implication in grid operational benefits.

4 Significant Accomplishments and Conclusions

The project team reports successful design of a novel probabilistic net-load forecasting architecture, comprising of a variational autoencoder and a recurrent neural network, which demonstrates 30% improvement in forecast performance, 60% improvement in training time, and consumes 44% less memory, when compared with conventional baseline models. The team tested the VRN3P model performance on GridLAB-D test-cases representing varying BTM solar penetration levels of 20%, 30%, and 50%, with integrated time-series net-load profiles provided by the utility partner (PGE). The VRN3P model demonstrate $<$2% hourly MAPE (averaged over the year) for day-ahead net-load forecast on the test scenario with 20% BTM solar. Transfer learning extension of the VRN3P model has demonstrated $8.37\times$ speed-up in training, while still achieving acceptable forecast performance of 2.24% hourly MAPE on the 30% BTM solar penetration test-scenario. Tested on the Solar Forecast Arbiter, the model achieved high CRPSS +0.3 and +0.08, compared to a given reference, in two locations with different solar penetration in TX (19%) and GA (63%), respectively. A preliminary version of the VRN3P GridAPPS-D™has been developed, along with a web-based interactive user-interface (named `Forte`) which has made available on GitHub for public use. In particular, `Forte` empowers users (grid operational planners) in gaining in-depth understanding of the forecast model's performance, effectively leveraging data visualization techniques to aid informed decision-making. It combines interactive visualization with performance metrics to instill greater trust in model outcomes, providing users with the flexibility to probe net-load predictions as a function of input variables like temperature and humidity. Finally, since most utilities (including PGE) do not use feeder-level net-load forecasts in operational decisions, the project team chose to measure VRN3P financial impact at the balancing authority (BA) level emulating participation in CAISO Energy Imbalance Market (Real Time Market) at the PGE trading node. Leveraging PNNL GRAF-Plan tool, a work-flow was constructed to demonstrate significant estimated reduction in balancing reserves procurement due to improvement in day-ahead net-load forecasts generated by VRN3P, when compared to CAISO 2020 forecast benchmark.

5 Path Forward

The project team will continue to explore opportunities for technology transfer and commercialization of the different products developed in this project. Through participation in various technical sessions, including the SETO AI/ML Workshop (Oct 2023), the project team has identified several small-business industries that remain interested in the VRN3P project outcomes, such as the AI/ML model architectures used for forecasting, or the possibility of extending the forecast models for dynamic hosting capacity estimation. In particular, the project team will explore opportunities within the DOE Technology Commercialization Fund (TCF) to transition the open-source licensed technologies from VRN3P in developing applications (such as hosting capacity estimation). Other future applications include the transition of the visual analytics-based interactive user-interface into an immersive human-AI environment for various AI/ML models.

6 Products

Software Release
Bhattacharjee K., S. Kundu, O. Vasios, A. M. Campbell, and A. Reiman. 2023. “FORTE: Interactive User-Interface Application for Deep Probabilistic Day-Ahead Net-Load Forecasting.” Link to GitHub repository: `https://github.com/pnnl/Forte`

Publications
Bhattacharjee K., S. Kundu, I. Chakraborty, and A. Dasgupta. 2025. “Who should I trust? A Visual Analytics Approach for Comparing Net Load Forecasting Models.” In IEEE PES Grid Edge Technologies Conference & Exposition (Grid Edge 2025), January 21-23, 2025, San Diego, CA. PNNL-SA-197642. doi:10.1109/GridEdge61154.2025.10887523
Campbell A.M., S. Kundu, A.P. Reiman, O. Vasios, I. Beil, and A. Eiden. 2024. “Clustering Interval Load with Weather to Create Scenarios of Behind-the-Meter Solar Penetration.” In IEEE Power & Energy Society General Meeting (PESGM 2024), July 21-25, 2024. PNNL-SA-192596. doi:10.1109/PESGM51994.2024.10688495
Bhattacharjee K., S. Kundu, I. Chakraborty, and A. Dasgupta. 2024. “Forte: An Interactive Visual Analytic Tool for Trust-Augmented Net-Load Forecasting.” In IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT 2024), February 19-22, 2024, Washington, D.C.. PNNL-SA-189754. doi:10.1109/ISGT59692.2024.10454191
Sen D., I. Chakraborty, S. Kundu, A.P. Reiman, I. Beil, and A. Eiden. 2022. “kPF-AE-LSTM: A Deep Probabilistic Model for Net-Load Forecasting in High Solar Scenarios.” arXiv. PNNL-SA-166731. doi:10.48550/arXiv.2203.04401

7 Project Team and Roles

Participant	**Organization**	**Role**
Soumya Kundu	PNNL	Principal investigator (PI), overall coordination
Indrasis Chakraborty	LLNL	Co-PI, lead AI/ML forecasting model developer
Andrew Reiman	PNNL	Lead AI/ML forecasting model validation
Allison Campbell	PNNL	Support model validation and grid impact analysis
Orestis Vasios	PNNL	Support synthetic test-case generation for validation
Kaustav Bhattacharjee	PNNL	Student intern, user-interface developer
Deepthi Sen	LLNL	Student intern, AI/ML model training
Andy Eiden	PGE	Support project with access to anonymized data
Ian Beil	PGE	Support project in an advisory role

8 References

[1] Ghadah Alkhayat and Rashid Mehmood. “A review and taxonomy of wind and solar energy forecasting methods based on deep learning”. In: *Energy and AI* (2021), p. 100060.

[2] Kaustav Bhattacharjee et al. “Forte: An Interactive Visual Analytic Tool for Trust-Augmented Net Load Forecasting”. In: *2024 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE. 2024, pp. 1–5.

[3] Kaustav Bhattacharjee et al. *pnnl/Forte*. Tech. rep. Pacific Northwest National Laboratory (PNNL), Richland, WA (United States), 2023.

[4] Kaustav Bhattacharjee et al. “Who should I trust? A Visual Analytics Approach for Comparing Net Load Forecasting Models”. In: *2025 IEEE PES Grid Edge Technologies Conference & Exposition (Grid Edge)*. IEEE. 2025, pp. 1–5.

[5] Charles Blundell et al. “Weight uncertainty in neural network”. In: *International Conference on Machine Learning*. PMLR. 2015, pp. 1613–1622.

[6] Allison M Campbell et al. “Clustering interval load with weather to create scenarios of behind-the-meter solar penetration”. In: *2024 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE. 2024, pp. 1–5.

[7] Junyoung Chung et al. “A recurrent latent variable model for sequential data”. In: *Advances in neural information processing systems* 28 (2015), pp. 2980–2988.

[8] Mingjian Cui et al. “Characterizing and analyzing ramping events in wind power, solar power, load, and netload”. In: *Renewable energy* 111 (2017), pp. 227–244.

[9] Aritra Dasgupta et al. “Familiarity Vs Trust: A Comparative Study of Domain Scientists’ Trust in Visual Analytics and Conventional Analysis Methods”. In: *IEEE trans. on visualization and computer graphics* 23.1 (2016), pp. 271–280.

[10] Timothy DelSole and Michael K Tippett. “Forecast Comparison Based on Random Walks”. In: *Monthly Weather Review* 144.2 (2016), pp. 615–626.

[11] Kate Doubleday, Vanessa Van Scyoc Hernandez, and Bri-Mathias Hodge. “Benchmark probabilistic solar forecasts: Characteristics and recommendations”. In: *Solar Energy* 206 (2020), pp. 52–67.

[12] Martin Gerlach, Francesc Font-Clos, and Eduardo G Altmann. “Similarity of symbol frequency distributions with heavy tails”. In: *Physical Review X* 6.2 (2016), p. 021009.

[13] Malini Ghosal et al. “Grid Reserve and Flexibility Planning Tool (GRAF-Plan) for Assessing Resource Balancing Capability under High Renewable Penetration”. In: *IEEE Open Access Journal of Power and Energy* (2022).

[14] Vinay Gupta et al. *Comprehensive review on effect of dust on solar photovoltaic system and mitigation techniques*. en. Oct. 2019. DOI: `10.1016/j.solener.2019.08.079`. URL: `http://dx.doi.org/10.1016/j.solener.2019.08.079`.

[15] Sue Ellen Haupt et al. “The SunCast solar-power forecasting system: the results of the public-private-academic partnership to advance solar power forecasting”. In: *National Center for Atmospheric Research (NCAR), Boulder (CO): Research Applications Laboratory, Weather Systems and Assessment Program (US)* (2016).

[16] Mohammad Safayet Hossain and Hisham Mahmood. “Short-term photovoltaic power forecasting using an LSTM neural network and synthetic weather forecast”. In: *IEEE Access* 8 (2020), pp. 172524–172533.

[17] Zhichun Huang, Rudrasis Chakraborty, and Vikas Singh. “Forward Operator Estimation in Generative Models with Kernel Transfer Operators”. In: *arXiv preprint arXiv:2112.00305* (2021).

[18] Weather Underground Inc. *Weather Underground, Inc.* URL: `https://lccn.loc.gov/2004564291` (visited on 11/08/2022).

[19] Akshith Reddy Kandakatla et al. “Towards trust-augmented visual analytics for data-driven energy modeling”. In: *2020 IEEE Workshop on TRust and EXpertise in Visual Analytics (TREX)*. IEEE. 2020, pp. 16–21.

[20] Amanpreet Kaur, Lukas Nonnenmacher, and Carlos F.M. Coimbra. "Net load forecasting for high renewable energy penetration grids". In: *Energy* 114 (2016), pp. 1073–1084. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2016.08.067. URL: https://www.sciencedirect.com/science/article/pii/S036054421631180X.

[21] Durk P Kingma et al. "Improved variational inference with inverse autoregressive flow". In: *Advances in neural information processing systems* 29 (2016), pp. 4743–4751.

[22] Stefan Klus, Ingmar Schuster, and Krikamol Muandet. "Eigendecompositions of transfer operators in reproducing kernel Hilbert spaces". In: *Journal of Nonlinear Science* 30.1 (2020), pp. 283–315.

[23] Soumya Kundu et al. *Developing and Validating Semi-Markov Occupancy Generative Models: A Technical Report*. 2021. arXiv: 2112.11111 [cs.LG].

[24] Soumya Kundu et al. "Optimal Energy Consumption Forecast for Grid Responsive Buildings: A Sensitivity Analysis". In: *2018 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE. 2018, pp. 230–236.

[25] Todd Levin et al. "Hydropower Value Drivers". In: (Feb. 2023). DOI: 10.2172/1991482. URL: https://www.osti.gov/biblio/1991482.

[26] Chao Liu et al. "Estimating Aggregated Behind-the-Meter Photovoltaic Generation Using Smart Meter Data". In: *2020 IEEE Power & Energy Society General Meeting (PESGM)*. 2020, pp. 1–5. DOI: 10.1109/PESGM41954.2020.9281564.

[27] Colleen Lueken, Gilbert E Cohen, and Jay Apt. "Costs of solar and wind power variability for reducing CO2 emissions". In: *Environmental science & technology* 46.17 (2012), pp. 9761–9767.

[28] Jian Luo, Tao Hong, and Shu-Cherng Fang. "Benchmarking robustness of load forecasting models under data integrity attacks". In: *International Journal of Forecasting* 34.1 (2018), pp. 89–104.

[29] Yuri V. Makarov et al. "Operational Impacts of Wind Generation on California Power Systems". In: *IEEE Transactions on Power Systems* 24.2 (2009), pp. 1039–1050. DOI: 10.1109/TPWRS.2009.2016364.

[30] Dieter H Mayer. "The Ruelle-Araki transfer operator in classical statistical mechanics". In: (1980).

[31] Ronald B Melton et al. *GridAPPS-D conceptual design V1. 0*. Tech. rep. Pacific Northwest National Lab.(PNNL), Richland, WA (United States), 2017.

[32] Frank A Monforte and Inc Itron. *Improving Short-term Load Forecasts by Incorporating Solar PV Generation: Interim Project Report*. California Energy Commission, 2017.

[33] Jaime Pizarroso, José Portela, and Antonio Muñoz. "NeuralSens: sensitivity analysis of neural networks". In: *arXiv preprint arXiv:2002.11423* (2020).

[34] K Porter et al. "Review of industry practice and experience in the integration of wind and solar generation". In: *PJM Renewable Integration Study: Task Report. GE Energy Consulting, New York* (2012).

[35] Xiangyun Qing and Yugang Niu. "Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM". In: *Energy* 148 (2018), pp. 461–468.

[36] Kevin P Schneider et al. "Modern Grid Initiative Distribution Taxonomy Final Report". In: (Nov. 2008). DOI: 10.2172/1040684. URL: https://www.osti.gov/biblio/1040684.

[37] Deepthi Sen et al. "KPF-AE-LSTM: A Deep Probabilistic Model for Net-Load Forecasting in High Solar Scenarios". In: *arXiv preprint arXiv:2203.04401* (2022).

[38] Masoud Sobhani, Tao Hong, and Claude Martin. "Temperature anomaly detection for electric load forecasting". In: *International Journal of Forecasting* 36.2 (2020), pp. 324–333.

[39] Dennis W Van der Meer, Joakim Widén, and Joakim Munkhammar. "Review on probabilistic forecasting of photovoltaic power production and electricity consumption". In: *Renewable and Sustainable Energy Reviews* 81 (2018), pp. 1484–1512.

[40] Fei Wang et al. "Wavelet decomposition and convolutional LSTM networks based improved deep learning model for solar irradiance forecasting". In: *applied sciences* 8.8 (2018), p. 1286.

[41] Yi Wang et al. "Probabilistic individual load forecasting using pinball loss guided LSTM". In: *Applied Energy* 235 (2019), pp. 10–20.

[42] Jason Yosinski et al. "How transferable are features in deep neural networks?" In: *Advances in neural information processing systems* 27 (2014).

[43] Yunjun Yu, Junfei Cao, and Jianyong Zhu. "An LSTM short-term solar irradiance forecasting under complicated weather conditions". In: *IEEE Access* 7 (2019), pp. 145651–145666.

[44] Hangxia Zhou et al. "Short-term photovoltaic power forecasting based on long short term memory neural network and attention mechanism". In: *IEEE Access* 7 (2019), pp. 78063–78074.

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354

1-888-375-PNNL (7665)

www.pnnl.gov