Received July 29, 2021, accepted August 4, 2021, date of publication August 6, 2021, date of current version August 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3103126

Regression and Generalized Additive Model to Enhance the Performance of Photovoltaic Power Ensemble Predictors

ADITYA SUNDARARAJAN, (Member, IEEE), AND BEN OLLIS, (Senior Member, IEEE)
Oak Ridge National Laboratory, Grid Components and Controls Group, Oak Ridge, TN 37932, USA

Corresponding author: Aditya Sundararajan (sundararajaa@ornl.gov)

This manuscript has been authored by UT-Battelle, LLC based upon work supported by the U.S. Department of Energy, Office of Electricity under contract number DE-AC05-00OR22725. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for U.S. Government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

ABSTRACT Photovoltaic (PV) power prediction has a constantly evolving solutions landscape with a myriad of data-driven techniques. Each technique leverages a self-adaptive algorithm that must retrain in intervals, be it each day, week, or season, to avoid the model generalizing poorly because of overfitting, underfitting, or concept drift. This paper aims to improve the generalization capability of PV power predictors such as autoencoders used widely in the industry by introducing feature-enhanced ensemble learning (FEEL) after the feature selection step. This framework uses a combination of nonparametric regression and generalized additive models, and an ensemble of weak regularized multilayer perceptron models. Once trained, the framework can reliably generalize on test data across long time periods without any significant degradation in performance. The proposed framework was validated against the baseline autoencoder-based feature enhancement model on a real PV system from a smart neighborhood in Alabama for September 2019. The FEEL framework performed three times better than the baseline, but when applied to the baseline, its performance improved by two times on average. Furthermore, the framework generalized consistently better than five other feature enhancement strategies. Despite fluctuations in weather, the FEEL framework's R-square score had a range of 8.1%, whereas that of the baseline was 48.3%. The mutual information and Minkowski distance scores attempted to quantify concept and model drift, respectively. These scores show that the FEEL framework generalized the ensemble learning models at least two times better than the baseline across the different test days. These results form the first step toward decentralized intelligence for smart grid applications that could free up resources for other expensive analytics in the field.

INDEX TERMS Feature enhancement, nonparametric regression, ensemble learning, generalized additive model, concept drift, photovoltaic power prediction.

NOMENCLATURE

Unless otherwise specified, the following nomenclature is applied throughout this paper.

AE	Autoencoder, also denotes one of the feature enhancement strategies
ARIMA	Autoregressive integrated moving average
ARIMAX	ARIMA with extragenous inputs
ERM	Ensemble of regularized multilayer perceptrons
FEEL	Feature-enhanced ensemble learning
GAM	Generalized additive model
LSTM	Long short-term memory
MAE	Mean absolute error
MLP	Multilayer perceptron
PV	Photovoltaic
PolyFit	Non-parametric regression with polynomial curve-fitting
PolyGAM	PolyFit used in conjunction with GAM
PolyGAMAE	PolyGAM used as inputs to autoencoder
RMSE	Root mean square error
SARIMA	Seasonal ARIMA
SARIMAX	Seasonal ARIMA with exogenous inputs
SSE	Sum of squared errors
SST	Total sum of squares

The associate editor coordinating the review of this manuscript and approving it for publication was N. Prabaharan.

Constants	
ϵ	a constant to adjust the partial dependency link function
N	the number of independent feature vectors
n	the length of a given feature vector
Functions	
$\mathbb{E}\{\cdot\}$	The expectation function
$\mathcal{L}(\cdot)$	The loss function used by the models in an ERM
$f_i(\cdot)$	The function that models nonlinear relationships between $x_i \in X$ and Y
$g(\cdot)$	The link function between the target and the feature vectors
$p(x_i, Y)$	Partial dependence between x_i and Y
Indices	
i	an index $\in [1, N]$,$\mathbb{Z}^+$ to denote a feature of interest
j	an index $\in [1, N] - \{i\}$ to denote features not of interest
Sets	
$\mathbb{Z}^+$	a set of positive integers
X	a multivariate set of N vectors in the input feature space FS
Y	a univariate set of the target vector Gen
Variables	
$\hat{X}$, $\hat{Y}$	Corresponding vectors of predicted X and Y values
μ	The mean of a given random variable
ρ_p	Pearson correlation coefficient
ρ_s	Spearman's Rho
σ	The standard deviation of a given random variable
AP	Air pressure, an input feature
AT	Ambient temperature, an input feature
FS	Input feature space
FS_A	Feature space updated using autoencoder
FS_G	Feature space updated using GAM
FS_P	Feature space updated using PolyFit
FS_{PGA}	Feature space updated using PolyGAMAE
FS_{PG}	Feature space updated using PolyGAM
G_x	Gini's mean difference for vector x
Gen	PV power output, the target variable
GI_x	Gini's index for vector x
I	Irradiance, an input feature
l	Neural network layer of a model in an ERM
p	Order of distance measure; $p = 1$ for Manhattan and $p = 2$ for Euclidean
R^2	R-square score
RH	Relative humidity, an input feature
WS	Wind speed, an input feature
Xr, Yr	ranked vectors corresponding to X and Y

I. INTRODUCTION

Prediction of photovoltaic (PV) power generation is a crucial component in monitoring system behavior and planning for on-demand dispatch [1]. Existing predictive techniques can be categorized as physics-based, model-driven, data-driven, or a combination thereof. Given how the generation curve follows the sun's diurnal cycles with intermittencies due to cloud cover and changing regional weather patterns, the prediction problem is challenging.

Recent works (elaborated in Section II) have demonstrated the statistical significance of the input feature space FS—which includes irradiance I (W/m^2), ambient temperature AT (°F), wind speed WS (m/s), air pressure AP (bars), and relative humidity RH (%) such that $FS = X = \{I, AT, WS, AP, RH\}$—on PV power Gen. Distribution grid networks are poised to adopt decentralized paradigms such as the Internet of Things–enabled Edge or Fog computing, which have allowed intelligence to partially shift from central cloud-driven data centers to the field [2]. The sensitivity of field data has also prompted the consideration of distributed learning frameworks that preserve privacy and ensure security by data abstraction. These emerging signs point to a future in which distributed energy resources such as PV, and even microgrids that encompass such distributed energy resources, will engage in localized predictions that can then be aggregated at the control center.

However, these developments must still address memory and compute resource constraints to perform specific operations. One such operation that falls within the scope of this paper is model retraining. Besides depending on additional resource consumption, the frequency of retraining a model depends on concept drift caused by changes in the data distributions over time, model drift due to poor generalization over time, data bias and variance, deployment and patching costs, and the dynamism of measurement domain.

Although adding feature engineering techniques increases computational burden, this paper aims to integrate two lightweight techniques: nonparametric regression (PolyFit) and generalized additive model (GAM) to capture both linear and nonlinear partial dependencies between FS and Gen. The generalization of the predictor should be significantly improved by integrating these techniques in the feature enhancement step. This claim is the first hypothesis of this paper, $\mathcal{H}_A$. Second, the availability of big data and the adoption of powerful machine learning libraries have prompted utilities to retrain their predictive models daily, weekly, or as often as each time FS is updated. However, by using an ensemble of "weak" models to make an ensemble of predictions—rather than a single predictor that makes point predictions—that can then be averaged, the improved generalization claimed by $\mathcal{H}_A$ can be sustained for longer test periods without having to retrain the models. This claim, framed as the second hypothesis, $\mathcal{H}_B$, is also validated by quantifying concept and model drift through distance similarity measures such as mutual information and Minkowski distances. By virtue of these hypotheses, the local resources can be freed up for more mission-critical analytics. For implementation, this paper uses data for the month of September 2019 from a smart neighborhood in Alabama.

The overarching goal of this work is to develop a data-driven framework that improves ($\mathcal{H}_A$) and sustains ($\mathcal{H}_B$) the generalization of PV power predictors along with exhibiting a reduced risk of overfitting. The terms "models" and "algorithms" are used interchangeably unless specified otherwise.

The key contributions of this work are that it.

1) Devises a feature-enhanced ensemble learning (FEEL) framework using PolyFit and GAM that captures linear and nonlinear partial dependencies in the data to improve the generalization of a day-ahead PV power predictor powered by an ensemble of regularized multilayer perceptrons (ERM) and accept or reject $\mathcal{H}_A$;
2) Analyzes the trends in model performance over multiple test data after being trained on a single training-validation set to determine if the improvement in generalization is also sustained, thereby accepting or rejecting $\mathcal{H}_B$;
3) Quantifies model drift by exploring the inherent relationships between the predictors in *FS* using pairwise correlations and between the actual and predicted values of *Gen* using normalized Minkowski distances;
4) Quantifies concept drift by exploring the mutual information between the training-validation set and each of the test days for the target vector *Gen*;
5) Delineates the partial linear and nonlinear dependencies between each predictor variable and the target to better profile the latent relationships between the parameters and exploit those relationships during the feature engineering stage
6) Compares the performance of FEEL framework with similar day-ahead PV power prediction techniques from the very recent literature; and
7) Provides a strategic roadmap for interested stakeholders to enable better allocation of PV plants for smarter operational planning and prioritizing resources for improved reliability and availability.

Six feature enhancement strategies were used, each labeled as a "case" and summarized in Table 1: **(a)** None: no feature enhancement, and *FS* is fed into the ERM; **(b)** AE: considered as the industry baseline, it consists of an autoencoder that takes *FS* as its input to yield the encoded-decoded output FS_A, which is fed into the ERM; **(c)** PolyFit: each vector from *FS* is fed into a nonparametric regressor to yield a new space FS_P that is fed into the ERM; **(d)** GAM: all vectors in *FS* are fed into a GAM modeler that generates Gen_G, which captures the partial dependencies between *FS* and *Gen*; the new feature space $FS_G = FS + Gen_G$ is fed into the ERM; **(e)** PolyGAM: a combination of PolyFit and GAM; $FS_{PG} = FS_P \cup Gen_G$; and **(f)** PolyGAMAE: a combination of AE and PolyGAM; FS_{PG} is used by AE to yield FS_{PGA} that is then fed into the ERM.

The rest of this paper is organized as follows. Section II summarizes the existing literature on the topic and highlights key shortcomings and challenges. Section III presents the proposed approach and formulated models. The results are presented in Section IV and discussed in Section V. Conclusions and future work are documented in Section VI.

TABLE 1. Cases based on the feature enhancement strategy used.

Case	Feature Space
None	$FS = X = \{I, WS, AT, RH, AP\}$
PolyFit	$FS_P = Regressor(FS)$
AE	$FS_A = Autoencoder(FS)$
GAM	$FS_G = FS \cup Gen_G$ where $Gen_G \leftarrow GAM(FS)$
PolyGAM	$FS_{PG} = FS_P \cup Gen_G$ where $Gen_G \leftarrow GAM(FS)$
PolyGAMAE	$FS_{PGA} = Autoencoder(FS_{PG})$

II. RELATED WORK

Different techniques in the literature address PV power prediction, which is interchangeably used with the term "forecasting." A simple numerical estimation approach combining historical data and system characteristics is proposed in [3], where the proposed method estimated values closer to the real observations in comparison to state-of-the-art estimators. Different variants of autoregressive integrated moving average (ARIMA) models such as ARIMA with exogenous inputs (ARIMAX), nonlinear autoregressive neural network with exogenous inputs, seasonal ARIMA (SARIMA), and SARIMA with exogenous inputs (SARIMAX) are used for forecasting and compared in [4], where the authors used forecast horizons of 1, 2, and 3 h. Mean absolute percentage error was used to assess performance. In another approach [5], the authors used SARIMAX, SARIMA, modified SARIMA, and neural networks for short-term forecasting. In this study, the power output horizon was classified into winter, spring, summer, and autumn. The authors conclude that neural networks, SARIMAX, and modified SARIMA were the best models to meet the forecasting needs for winter, spring, and autumn, whereas SARIMA performs better in summer. ARIMA was used to forecast global horizontal irradiance [6], the assessment of which is done by measuring the root mean square error (RMSE) and a coefficient of determination such as R^2. The best model recommended by this work had an R^2 of 0.89 and an RMSE of 0.729. In [7], a day-ahead solar forecasting was done using a support vector machine with different kernels, among which the radial basis function performed best with an RMSE of 0.580 and a mean absolute error (MAE) of 0.728. Neural networks were used by [8] to forecast solar irradiance, combined with a genetic algorithm to find the optimal array size and position of the solar monitoring station to get the most accurate forecast. Numerical weather prediction and convolutional neural networks have also been used to forecast solar energy [9]. In another study, the probabilistic forecasting of solar power was done using multiple linear regression [10]. Short-term forecasting using the Mycielski algorithm was done in [11] where a 60 W solar panel was used. Authors in [12] forecast irradiance using cloud motion and numerical weather prediction.

The work in [13] proposes to adjust the hyperparameters of long short-term memory (LSTM) PV power predictors to account for weight estimation and learning rate acceleration. However, additional techniques are required to counter the added computational complexity of their approach for online forecasting. Extraction of latent features from observed (historical) and forecast weather values was explored in a recent study [14], which made day-ahead predictions similar to this paper, but the proposed work differs from the cited study by capturing the linear and nonlinear partial dependencies between the predictors and the target. Another relevant study modeled the linear and nonlinear relationships between weather parameters and PV power using linear and ranked correlations [15]. Although correlations capture the nature of relationships between these variables, the causal dependencies between them are not captured. The proposed work adopts nonparametric regression for this purpose. A similar study aimed to apply feature selection using correlations and backward elimination [16], but it used linear regression to capture what could entail even nonlinear relationship between the variables. Deep learning architectures were applied to improve the performance of PV power prediction in [16]–[18] and in [19] for short-term electricity price forecasting, but the computation complexity of deep networks against traditional networks with enhanced features poses practical limits.

Several improvisations to long-term PV forecasting have emerged. A feature-selective ensemble learning for months-ahead forecast horizon in [16] compared the performances of simple mean, LSTM, convolutional neural networks, and their ensembles. The ensembles showed a consistently better performance than the individual models for up to a horizon of 10 months ahead. The research in [20] combined the selective point forecasts of autoregressive conditional heteroscedastic, multiple linear regression, and random forests to improve forecasting accuracy. Unlike the proposed work that averages ensemble predictions, the authors of that work employed a probabilistic prediction mechanism using quantile regressors. The uncertainties quantified therein helped better interpret the model's results. Hence, this paper's work will be extended to include quantile regression mechanisms for more explainable results.

The scopes of the following works with short-term prediction horizons align with that of the proposed FEEL framework. Therefore, the discussion in Section V will address them. Reference [21] divided the target variable into its regular and irregular components and used ARIMA and improved deep belief network to independently predict them for short-term horizons. The proposed FEEL framework follows a similar process, but it captures linear and nonlinear partial dependencies and uses them to modify the feature space. The dynamic nature of weather and its influence in the generated power was analyzed in [22]. The work exploited deep learning techniques to extract the hidden sequential patterns in PV power output using only the data captured around sunrise and not any further meteorological data. The authors in [23] also proposed an ensemble learning method that combined deep learning with time correlation principles. The ensemble framework was then evaluated against its components independently. The authors of [24] proposed an ensemble of autoencoder and LSTM to make short-term day-ahead power predictions and compare the results with other benchmarks such as persistent models and simple feedforward neural networks for up to 1 week ahead. Three other works [25]–[27] that proposed similar ensemble approaches using deep learning for short-term (mostly day-ahead) PV power forecasting are also included in the comparative study.

III. PROPOSED APPROACH AND MODEL FORMULATION

Figure 1 shows a comprehensive overview of the FEEL framework, its models, and how they interact with each other. The following subsections will discuss and formulate each of these models in detail. The section ends with the definition of performance evaluation metrics: RMSE, MSE, MAE, and R^2 for ensemble-to-ensemble comparisons [28], mutual information score for evaluating model drift [29], [30], and Minkowksi distances for evaluating concept drift [31].

A. DATA PREPARATION

The system considered for framework validation is a 330 kW PV power plant in an Alabama smart neighborhood that partially supports 62 homes as part of a community microgrid [32]. Data from the PV system are subject to exploratory analyses and cleaning as outlined in the authors' prior work for similarly sized PV systems [33]–[37]. These include imputing missing values [38] and statistical curve fitting [39]–[41]. Time-series data on $X = \{I, WS, AT, RH, AP\}$, and $Y = Gen$ from the month of September 2019 were used for this study. These data were recorded in 0.5-s intervals from multiple plant sensors. System-level measurements on plant voltage, current, power, frequency, and power factor were recorded by a revenue-grade meter at the plant's main switchgear. Additionally, regional weather data were gathered from the weather station at the Birmingham-Shuttlesworth Airport. The data points were available in hourly intervals and were used to determine the type of a given day (fair, cloudy, or rainy). In all the analyses conducted hereon, the completeness and integrity of the data were ensured.

B. AUTOENCODER AS THE BASELINE

Autoencoders are lossy data compression models that use self-supervised learning and comprise encoder and decoder layers built using neural networks. These networks are typically single layers of neurons that match the input dimensions but are separated by a latent space that represents the compressed version of the feature space. Their ability to effectively capture nonlinear dependencies inherent in the data and to represent original feature space despite the noise and aberrations in a compressed data set make them viable in the industry for feature enhancement or

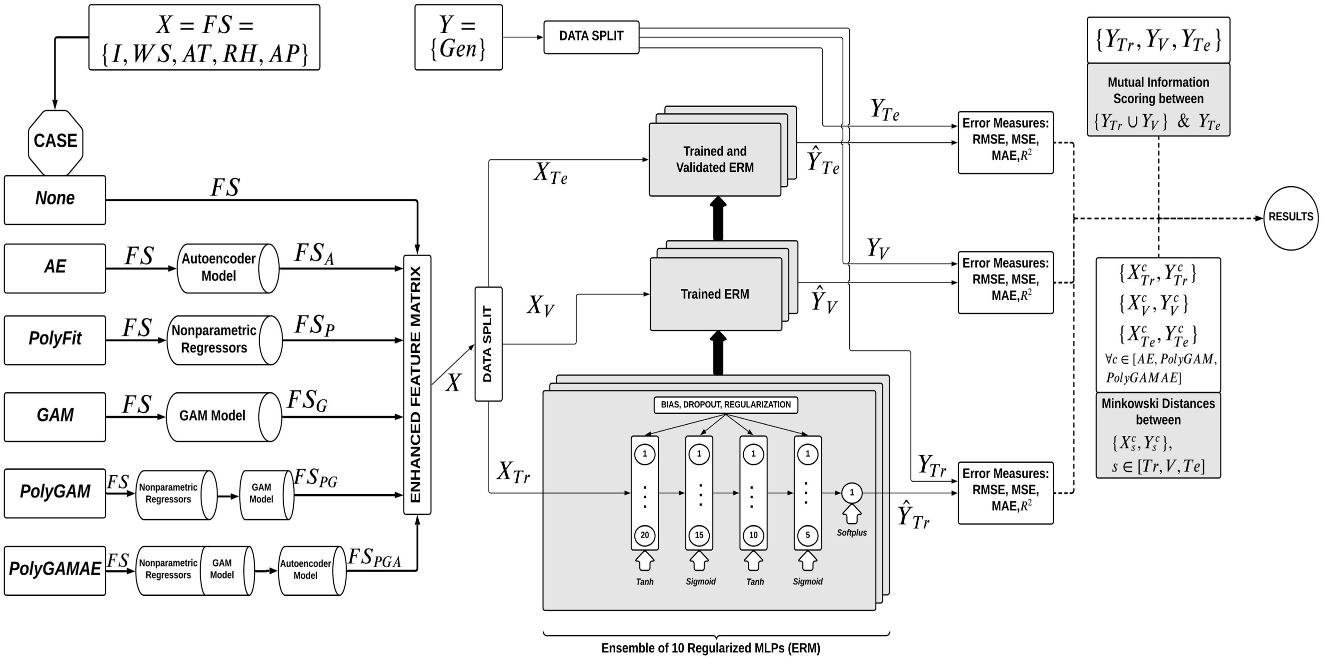


FIGURE 1. Flowchart summarizing the proposed approach for feature enhanced prediction.

for prediction itself. Recent advancements have resulted in the emergence of different variations to the original model, such as denoising and contractive autoencoders for feature extraction, and variational autoencoders for cases in which a greater control over the latent space is required. Typically, the AE model comprises an encoder and a decoder with the latent space between them. This work uses AE as a feature-enhancement step. The encoder compresses the FS to generate the encoded latent space, which is then taken by the decoder to generate FS_A that carries forward the useful representations and dependencies in FS. Given the prominence of fully connected autoencoders in the literature for this use [42]–[44], [44]–[46], an autoencoder was considered as a baseline model in this paper and used in two cases (AE and PolyGAMAE) as shown in Table 1. Algorithm 1 summarizes the model.

Algorithm 1 Autoencoder for Cases AE and PolyGAMAE to Respectively Generate FS_A and FS_{PGA}

```
Inputs: $\{X_{Tr}, X_V, X_Te\}$ from $FS$; $Dim_{FS}$, dimension of $FS$; $Dim_{EN}$, dimension of latent space
Define AE with layers $[L_1 := Dim_{FS}, L_2 := Dim_{EN}, L_3 := Dim_{FS}]$
Define AE with activations [Between $L_1$ and $L_2$ := $Tanh$, between $L_2$ and $L_3$ := $Tanh$]
Set parameters $P$ := [Optimizer, epochs, loss function]
Compile AE with $P$
Train AE with $X_{Tr}$ and update AE
Validate AE with $X_V$
$FS_A := \text{AE}(X_{Te})$
Save AE for future use
Return $FS_A$ to be used for ensemble predictions
```

C. NONPARAMETRIC REGRESSION (POLYFIT)

Unlike parametric regression that exploits a known, predetermined relationship, nonparametric regression discovers the relationships through curve-fits that minimize a prescribed error measure, RMSE in this case. Different models exist, such as local averaging [47], local regression [14], kernel smoothing [48], [49], and wavelet transforms [50]–[53]. The proposed approach summarized in Algorithm 2 applies polynomial fitting with up to 9° to capture dependencies, and the degree with the lowest error score is used as the best fit. These minimized scores are shown in Table 3.

D. GAM

GAM has been applied on time-series data for forecasting [54]–[56], which can be represented as semi-parametric functions of the form $g(\mathbb{E}\{Y|X\}) \leftarrow \epsilon + \sum_{i=1}^{N} f_i(X_i)$, where X_i, $i \in [1, N], \mathbb{Z}^+$ is a set of N independent variables in $X \in FS$ linked to Y, the dependent target variable, through the link function $g(\cdot)$. ϵ captures a constant. The set of feature functions $f_i(\cdot)$ model the nonlinear relationships between X and Y from B splines.

This stage also conducts a partial dependence analysis to explore the influence of each vector in X on Gen with a confidence interval of 95% [57]. The influence can be monotonic, linear, or nonlinear. If FS denotes the input feature space that comprises X_i, the feature of interest, then X_j denotes the features not of interest such that $X_i \cup X_j = FS$. A partial dependence function $p(X_i, Y)$ captures the causal relationship between X_i and the target Y, which

Algorithm 2 Nonparametric Regression for Cases PolyFit, PolyGAM, and PolyGAMAE to Respectively Generate FS_P, FS_{PG}, and FS_{PGA}

```
Inputs: $\{X_{Tr}, X_V, X_{Te}\}$ from $FS$; $\{Y_{Tr}\}$ from $Gen$; $D := [1, 9]$, the set denoting the degree search space
for $i$ in $1 : D$ do
    Fit a least squares polynomial on $[X_{Tr}, Y_{Tr}, i]$
    Set $W :=$ weights from the fitted polynomial
    Build the function PF s.t. $Y_{Tr} \leftarrow \mathrm{PF}(X_{Tr}, W)$
    $\hat{X}_{Tr} \leftarrow \mathrm{PF}(X_{Tr})$
    Compute $RMSE_i$ between $\hat{X}_{Tr}$ and $\mathrm{PF}(X_{Tr})$
Set optimal degree, $d :=$ degree $\in D$ with min$(RMSE)$
Set PF := function with weights mapped to the degree $d$
Validate PF with $X_V$
$FS_P := \mathrm{PF}(X_{Te})$
Return $FS_P$ to be used for ensemble predictions
```

is marginalized over the set X_j. In doing so, the average marginal effect of a given feature on the target can be intuitively understood. This study, however, assumes little or no correlation between X_i and X_j.

E. ERM TO PREDICT PV POWER

Prediction models can be designed to deliver results in point predictions, ensemble predictions, or quantile predictions. Point predictions are subject to the inconsistencies and randomness inherent in the measurement domain, the data acquisition process, and the randomness in the models during the intialization of parameters. Ensemble predictions employ ''weak'' models in parallel to train and validate on the same or subsets of training and validation data. The individual predictions of these weak models are either banded together within a confidence interval or consolidated into point predictions using simple or weighted mean. Compared with point predictions, ensemble predictions better account for the model randomness and other inconsistencies. Therefore, this study employed an ensemble of 10 weak regularized multilayer perceptrons (MLPs), all of the same architecture training and validating the same sets of data. A simple mean of the results are regarded as the final predictions. Quantile predictions, which are better capable of capturing nonlinear dependencies between predictors and the target, are out of this work's current scope but will be explored in the future.

The ERM was designed to predict PV power *Gen* using a range of features that were dictated by the feature enhancement case as seen in Fig. 1. Depending on the case, five or six dimensions of features were considered. The ERM's architecture was designed with a combination of hyperparameters that yield the most accurate predictions. These parameters are summarized in Table 4. Each model in the ensemble comprises 1 input layer representing features *FS*, 4 hidden layers with 20, 15, 10, and 5 units, and an output layer with 1 unit, the power *Gen*. Each layer, l, has a specific activation function, with weights initialized to samples drawn from Xavier uniform distribution [58]: tanh activation [59] for l_2 and l_4, sigmoid [60] for l_3 and l_5, and softplus [61] for l_6. Mean square error (MSE) is used as the loss function, defined as $\mathcal{L}(\hat{Y}^{(l)}, Y^{(l)}) = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i^{(l)} - Y_i^{(l)})^2$, where $l \in \{l_1, l_2, l_3, l_4, l_5, l_6\}$, n is the number of samples, Y is the vector of observed values, and $\hat{Y}$ is that of the predicted values. Adaptive stochastic gradient with momentum (Adam) improves the traditional stochastic gradient descent by calculating individual adaptive learning rates from the first and second moment estimates of the gradients for each parameter [62]. Hence, it was used to train and validate the ensemble, and the resulting ensemble was reused during the testing phase as shown in Algorithm 3.

Algorithm 3 ERM Model to Predict *Gen* Given a Feature Vector Matrix *FS*

```
Inputs: $X$ denoting $FS$; $Y$ denoting $Gen$; $P$ denoting the model parameters
Split $X$ into $[X_{Tr}, X_V, X_{Te}]$, $Y$ into $[Y_{Tr}, Y_V, Y_{Te}]$
Set layers $L := P.\ L_i, i \in [1, 6], \mathcal{Z}^+$
Set dropouts $D := P.\ D_{ij}, i \in [1, 5], j \in [2, 6], i \neq j, (i, j)\mathbb{Z}^+$
Set activations $A := P.\ A_{ij}, i, j$ defined as above
Set Ensembles, $E := 10$; loss function $LF \leftarrow \mathrm{MSE}(\cdot)$
Build model ERM $\leftarrow \mathrm{MLP}(L, D, A)$
Build forward propagation function FProp(MLP)
for $e$ in $1 : E$ do
    for each epoch in 1: 150 do
        $\hat{Y}_{Tr}^{e} := \mathrm{ERM}(X_{Tr})$
        loss $:= \mathrm{LF}(\hat{Y}_e, Y)$
        Back propagate ERM$(\cdot)$ to update weights
        $\hat{Y}_{V}^{e} := \mathrm{ERM}(X_V)$
        Save ERM$_e$ for future use
        $\hat{Y}_{Te}^{e} := \mathrm{ERM}(X_{Te})$
    Compute error scores for training, validation, and testing sets
Return ERM$_i$, $i \in [1, 6], \mathbb{Z}^+$; error scores
```

Algorithm 4 Distance Scores to Quantify Model Drift

```
Input: $X$ and $Y$, the two vectors of variables that are to be compared
Define $n$, the total number of data points in $X$ and $Y$
Define $k \in [1, n]$
Normalize $X$ and $Y$ to the range $[0, 1]$
Set matrix minkist $:= [0]_{1,6}$
for each case $c$ in [AE, PolyGAM, PolyGAMAE] do
    manhattan-dist$[c] := [\sum_{k=1}^{n}(|X_k^c - Y_k^c|)]$
    euclidean-dist$[c] := \sqrt{[\sum_{k=1}^{n}(|X_k^c - Y_k^c|)^2]}$
Return manhattan-dist and euclidean-dist
```

F. EVALUATING DATA SIMILARITY

RMSE, MSE, and MAE were used as defined in [28]. R^2 scores, which determine the statistical closeness of the fitted predictions $\hat{Y}$ to the real observations Y, each of length n,

TABLE 2. A summary of the high-level statistics on each predictor variable and the target variable from the raw input feature set *FS*.

Variable	Minimum	1st Quantile	Median	Mean	3rd Quantile	Maximum	Gini's Mean Difference (G_X)	Gini's Index (GI_X)	Spearman's Rho (ρ_s)
I	2.995	202.866	509.909	521.709	861.707	1327.454	402.3	0.395	0.977914
WS	0	0.4384	1.5344	1.9503	2.8496	19.728	1.992	0.511	0.3626866
AT	15.73	28.3	31.75	30.72	34.04	39.51	4.931	0.078	0.398369
RH	23.39	40.56	47.46	52.55	60.72	100	18.1	0.191	−0.3570418
AP	1,141	1,146	1,148	1,149	1,151	1,157	3.713	0.002	0.06297039
Gen	1	55	160	161	270	328	125.8	0.391	1

were computed using Eq. (1), where SSE denotes the sum of squared errors.

$$SSE = \sum_{i}^{n}(Y_i - \hat{Y}_i)^2, \quad SST = \sum_{i}^{n}(Y_i - \bar{Y})^2,$$
$$R^2 = 1 - \frac{SSE}{SST} \tag{1}$$

A general representation of the Minkowski distance between two vectors X and Y of length n each is given by minkdist $= (\sum_{i=1}^{n}(|X_i - Y_i|)^p)^{1/p}$ where $p \geq 1$. In this work, Manhattan and Euclidean distances (with $p = 1$ and $p = 2$, respectively) are used. The usage of the two distance measures is further summarized in Algorithm 4, wherein both distance measures are calculated for each of the six cases and averaged across all sets (training, validation, and testing). These two distances were also normalized to ensure side-by-side comparison across the cases. The larger the values of these distances, the more dissimilar the data. By computing these distances between observed and predicted values for different sets, model drift is quantified.

Another metric, mutual information $MI_{X,Y}$, computes the statistical dependence between two random variables X and Y as a mutual information score as defined in information theory. This metric was used to quantify concept drift of the raw input data, where X is the training set and Y denotes the testing set from a specific date.

IV. RESULTS

This section presents results from the execution of the FEEL framework on a training set spanning 4 days from September 1 through 4, a validation set spanning 3 days from September 5 through 7, and testing on the rest of the month, all from 2019.

A. DATA PREPARATION

The complete process of model building and execution was conducted on a single 64-bit machine with 16 GB RAM powered by an x64-based i7 processor. All scripts were built on Python 3.9 with a PyCharm integrated development environment editor. A pairwise linear correlation analysis was conducted on the *FS* to determine any inter-feature correlations that might negate the assumption of mutual independence by the partial dependence computed by GAM in Stage 4. The correlation matrix, computed using Eq. (2) from [63] is visualized in Fig. 2. As expected, there is a strong positive correlation of 0.98 between I and Gen because both follow a diurnal trend. However, the relationships between other feature vectors and Gen vary as expected. For instance, there is a strong correlation of −0.86 between AT and RH, implying that a rising trend in relative humidity exists alongside a dropping trend in ambient temperature, which is indicative of rain conditions. Similarly, RH is negatively correlated with Gen. A correlation coefficient of 0.07 between AP and Gen indicates that these variables are nearly independent of each other. Therefore, the ERM's predictive performance would not change significantly if AP were to be removed from FS. Except AT and RH, no other pair of vectors within the feature space exhibit strong linear correlations, which implies that FS can be fit within the general assumption of GAM's partial dependence computations discussed in Section IV-D.

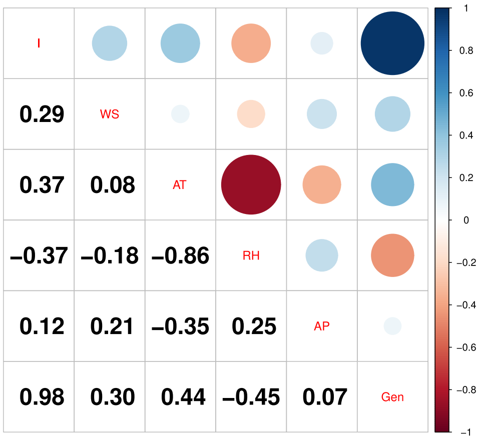
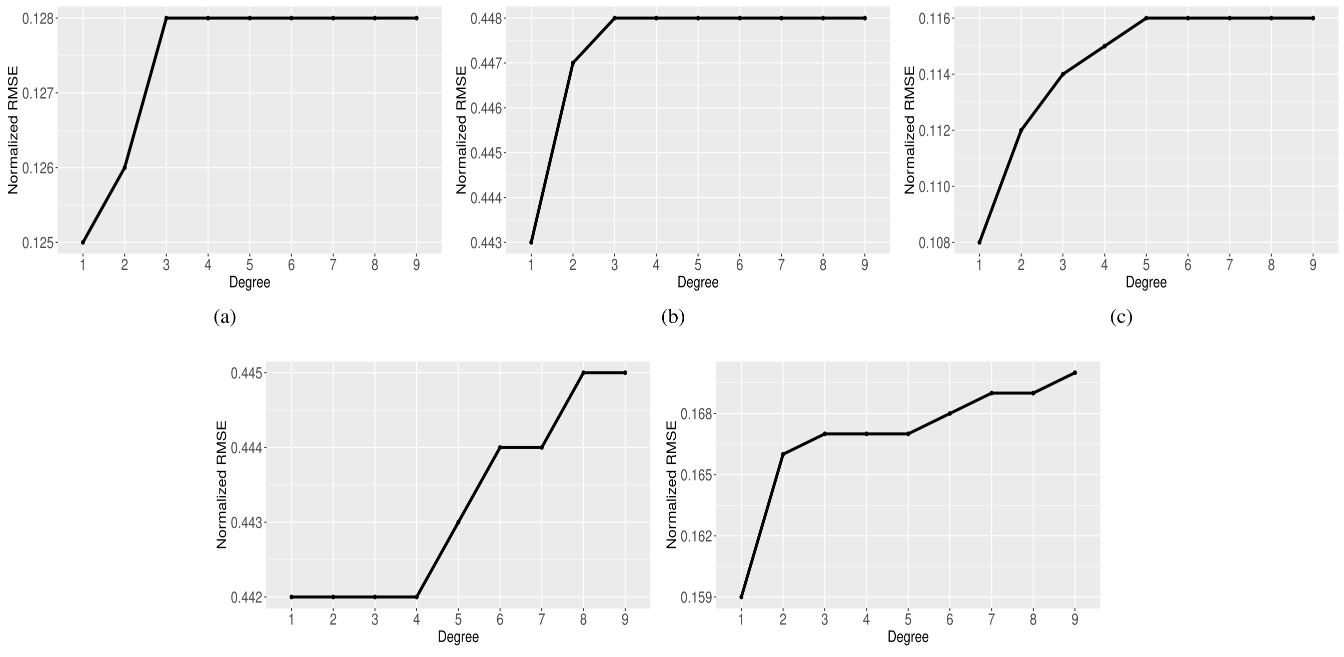


FIGURE 2. Pairwise Pearson correlation to explore linear relationships among *FS* and with *Gen*.

$$\rho_p = \frac{n(\sum X_i Y_i) + \sum X_i \sum Y_i}{\sqrt{n\sum X_i^2 - (\sum X_i)^2}\sqrt{n\sum Y_i^2 - (\sum Y_i)^2}} \tag{2}$$

Table 2 shows the standard statistics on each input vector in X as well as the target vector Gen over the entire data set. Two parameters in this exploratory analysis are of specific interest. Gini's mean difference (GMD) is computed using Eq. (3) from [64], where G_X denotes the mean difference for the vector X with n observations. GMD measures the



(a)

(b)

(c)

(d)

(e)

FIGURE 3. The trends in (a) RMSE, (b) MSE, (c) MAE, and (d) R-square values for the ERM across test days (September 8–30, 2019) categorized into cloudy (red circle), fair/sunny (green triangle), and rainy (blue square) days. The trends denote a periodicity in performance irrespective of the feature enhancement strategy.

inequality in a given vector of distribution. This metric is considered superior to variance as a measure of inequalities in a non-normal distribution [65], [66]. Gini index, GI_X, calculated from G_X by Eq. (3) better captures the inequality within a vector X. Unlike GMD, Gini index has a range of $[0, 1]$, and a larger index implies greater inequality. Table 2 shows that wide-ranging attributes such as I and Gen have higher Gini indices. Furthermore, their indices are comparable given the similarity between their distributions. Predictors with a concise range such as AT and AP have indices closer to 0, where an index of 0 implies no inequality. The variety of spread in data observations can be construed as a result of the inherent intermittency in the weather, and thus power output. The second parameter of interest is Spearman's Rho, ρ_s, computed using Eq. (4) from [67]. Unlike pairwise correlations that use raw data, rho_s uses ranked vectors Xr and Yr corresponding to the raw vectors X and Y given the repeating observations. ρ_s measures the monotonic relationships between the variables and has a range of $[-1, 1]$ like ρ_p, which measures linear relationships as shown in Eq. (4).

$$\begin{aligned} G_X &= \sum_{i=1}^{n} 4 \times \frac{i-(n-1)}{2n(n-1)} \times sort(X_i - mean(X)) \\ GI_X &= \frac{G_X}{2\mu} \qquad (3) \\ \rho_s &= \frac{n(\sum Xr_i Yr_i) + \sum Xr_i \sum Yr_i}{\sqrt{n\sum Xr_i^2 - (\sum Xr_i)^2}\sqrt{n\sum Yr_i^2 - (\sum Yr_i)^2}} \qquad (4) \end{aligned}$$

B. AUTOENCODER AS THE BASELINE

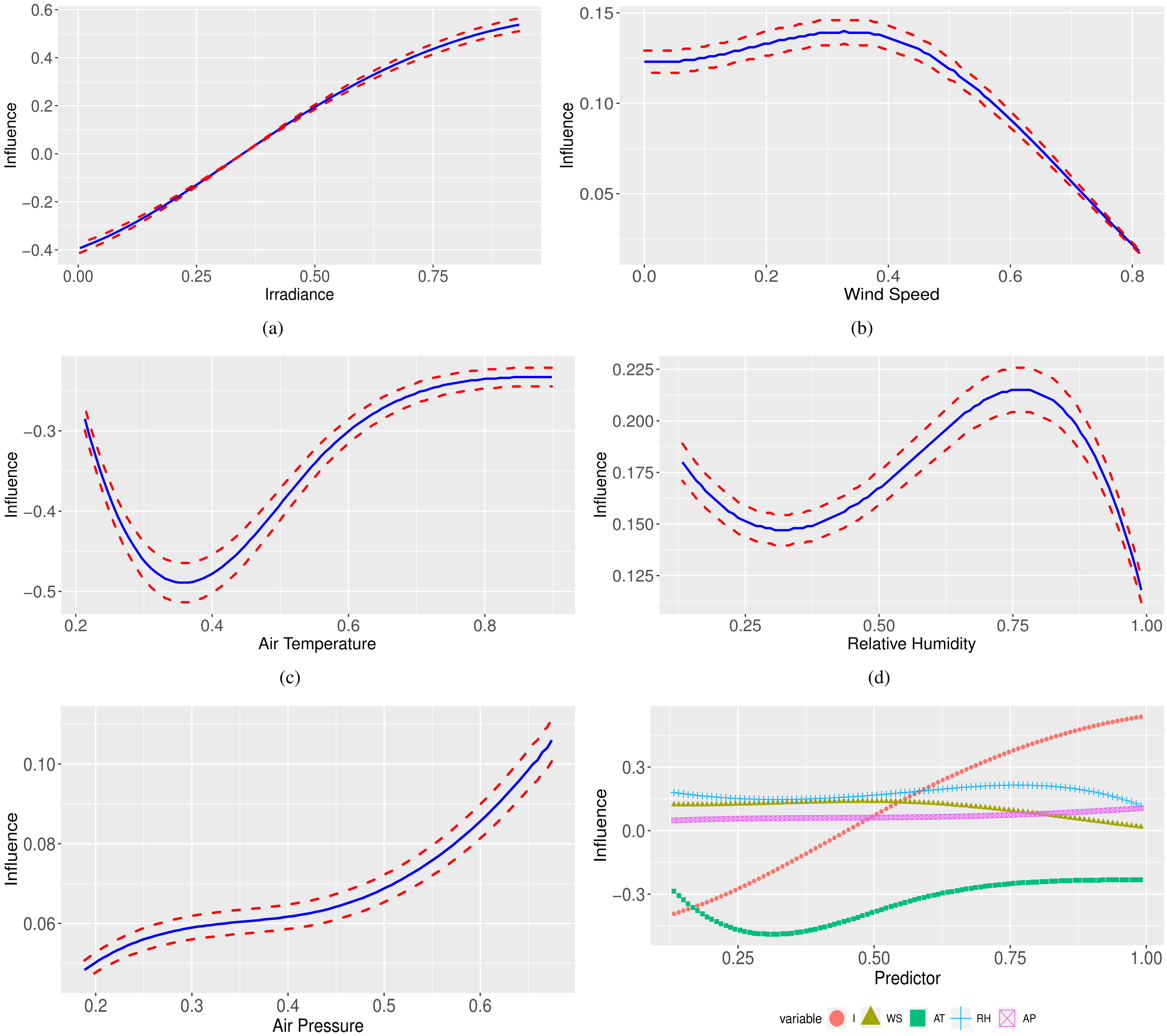
Table 3 summarizes error measures for the optimal AE model selected after running a grid search of its hyperparameters. The same AE model architecture, with its hyperparameters, was used for both feature engineering cases, AE and PolyGAMAE. The only difference between these cases is in the input data fed into the encoder part of the model.

C. NONPARAMETRIC REGRESSION

Table 3 shows RMSE values of the optimal PolyFit models of first order. Higher-order polynomials for each vector in X with Gen yielded larger RMSE values as illustrated by Fig. 3. Given the nonparametric regressors capture hidden relationships between features and the target, first order polynomials were fitted to yield corresponding feature space FS_P comprising five regressor features. The same feature space FS_P is used in cases PolyFit, PolyGAM, and PolyGAMAE to ensure the results are comparable.

D. GAM

Figure 4 shows how each vector in X, normalized to $[0, 1]$, affects Gen through the partial dependence plots. These plots can be studied in conjunction with the linear correlation plot in Fig. 2. The red dashed lines represent the confidence interval of the partial dependencies, and Fig. 4a shows that I has the strongest influence on Gen and with very high confidence. This finding corresponds with the high



(a) (b) (c) (d) (e) (f)

FIGURE 4. The partial dependence trends between each predictor variable (a) *I*, (b) *WS*, (c) *AT*, (d) *RH*, and (e) *AP* and the target variable *Gen*. Subfigure (f) shows all the curves side-by-side for easier comparison of relative trends.

TABLE 3. Error measures for optimal AE and PolyFit models.

Model	Measure	Feature	Value
Autoencoder	MSE	Baseline (case AE)	0.0121
		With PolyGAM (case PolyGAMAE)	0.0106
PolyFit	RMSE	I	0.1249
		WS	0.4429
		AT	0.1081
		RH	0.4417
		AP	0.1591

positive correlation between I and Gen. Parameters such as wind speed and air temperature are influenced indirectly by irradiance. Higher wind speeds result in faster-moving clouds, which can cause greater intermittencies in generation. Higher irradiance might cause a rise in air temperature, but PV module efficiency tends to lower at high temperature ranges. This effect is shown in Fig. 4c as a plateauing of the influence score. Air pressure has the widest confidence interval among all dependencies given its very small range of values as shown in Table 2. The general trend as shown in Fig. 4e demonstrates a positive partial dependence between each predictor and the target variable.

The slightly negative correlation between RH and Gen is shown more clearly in Fig. 4f, where an optimal value of RH is shown to have the greatest influence on Gen. In other words, very low relative humidity corresponds with lower power output, but the influence score increases proportionally until an optimal point. Beyond this value, PV power continues

TABLE 4. Parameters used in the FEEL framework modeling after grid search.

Model	Parameter	Value
AE	Optimizer	Adam
	Loss function	MSE
	Activation	Tanh
	Latent space dimension	$[1, 1]$
	Epochs	30
	Batch size	124
PolyFit	Degree search space	[1,9]
GAM	λ search space	$[rand(\mathrm{N}(0,1))]_{100\times5}$
	Splines	5
	$[\lambda_I, \lambda_{WS}, \lambda_{AT}, \lambda_{RH}, \lambda_{AP}]$	[2.649, 1.536, 1.02, 1.15, 1.448]
ERM	Layers	$[FS, 20, 15, 5, Gen]$
	Activations	$[Tanh, Sigmoid, Tanh, Sigmoid, Softplus]$
	Dropouts	[0.6, 0.5, 0.5, 0.5, 0.6]
	Loss function	MSE
	Epochs	150
	Optimizer	Adam
	Learning rate	0.1
	Regularizer	0.0005
	Ensemble size	10

TABLE 5. Execution time of the different feature enhancement modes subject to the system characteristics defined in section IV-A.

Model	Unit Time (s)	Description	Total Time (s)
PolyFit	7.732	Time to generate and save FS_P for PolyFit, PolyGAM, and PolyGAMAE	3×7.732 $= 23$
GAM	600	Time taken by GAM to generate FS_G and save the model for GAM, PolyGAM, and PolyGAMAE	3×600 $= 1,800$
AE	74.499	Time taken by AE to generate and save FS_A for AE and PolyGAMAE	$2 \times$ $74.499 =$ 149
Each ERM run	75	Time taken for each weak model in the ensemble of 10 to train and validate on a given feature space for each of the 6 cases	$10 \times 6 \times$ $75 =$ $4,500$
PolyFit	0 (approx.)	On the test data, FS_P is generated almost instantly for all 3 cases	0 (approx.)
GAM	0 (approx.)	On the test data, FS_G is generated almost instantly for all 3 cases	0 (approx.)
AE	37.945	Time taken by AE to load the saved model and regenerate FS_A for test data	$2 \times$ $37.945 =$ 76
Each ERM run	0.6	Time taken for each weak model in the ensemble of 10 to make predictions on a given test data's feature space for each of the 6 cases	$10 \times 6 \times$ $0.6 = 36$
Total Train and Validation Time			108 min
Total Test Time			113 s

to decrease as the relative humidity rises (which might be linked with rainy weather).

E. ERM TO PREDICT PV POWER

As discussed in Section III-E, an ensemble approach to predictions yields better averaged results than point predictions that could be influenced by the randomness in the initializers used by the models, and by the inherent system settings. Table 5 summarizes the execution times of the different feature enhancement modes. Training and validation take longer than generalization. However, when this information is viewed in conjunction with the ensemble prediction results in Table 6, it is clear that despite an increase in computation steps, the net overhead in execution time is not high.

The anticipated trade-off between generalization and retraining is overfitting and underfitting. Overfitting is caused by a high degree of variance, wherein the model performs well on training set but generalizes poorly on the validation and testing sets. Underfitting is caused by high inherent bias, where the model has not learned the full trends in the feature space and hence, makes poor predictions on the testing set. Figure 5 shows the variation of the loss function during the training and validation of the ensemble models for 150 epochs, averaged across the 10 of them for easier visualization. The close agreement between the losses during training and validation indicate a low degree of overfitting. Furthermore, across all six cases, the validation loss per epoch is slightly lower than the corresponding training loss, which is indicative of a good model fit. The loss curves also tend to stabilize over the epochs without any continuing significant decline, indicating that the model has sufficiently learned the inherent trends and thus, the risk of underfitting is low.

Table 6 summarizes errors for the six feature enhancement cases. The model is trained on days 1 through 4 and validated on days 5 through 7 of September 2019. The results are shown for specific types of data observed within the entire data set. First, errors for the training and validation sets are presented. The case that yields the best results is shown in bold-faced text for each data set. A consistent pattern emerges where AE, this study's baseline, performs poorly compared to the other cases. PolyFit and GAM individually perform decently, with the variance among their errors fairly low despite the trends in data such as varying weather. As stated in Section I, $\mathcal{H}_A$ is that PolyGAM shows a significant improvement in the generalization of ERM compared to the baseline AE. From these results, the consistency in PolyGAM's performance even in unfavorable weather conditions validates the hypothesis. In addition, PolyGAM also improves the performance of AE when combined, as seen in the demonstrable improvement from AE to PolyGAMAE.

The bar chart in Fig. 6 is similar to Fig. 5, but in Fig. 6, the bar charts show the variations in the average RMSE, MSE, and MAE scores computed between the actual and predicted PV power values during the training (the errors

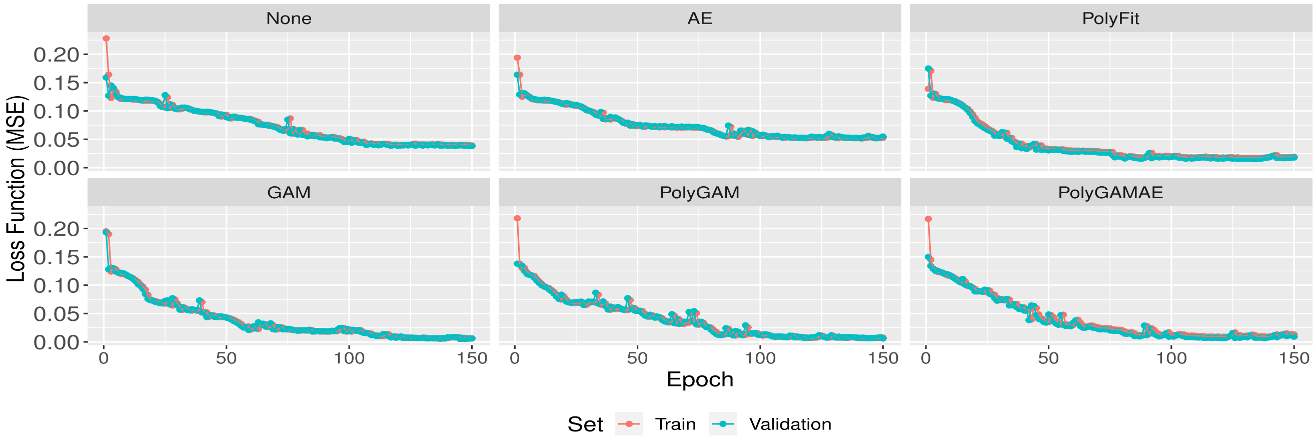


FIGURE 5. Flowchart summarizing the proposed approach for feature enhanced prediction.

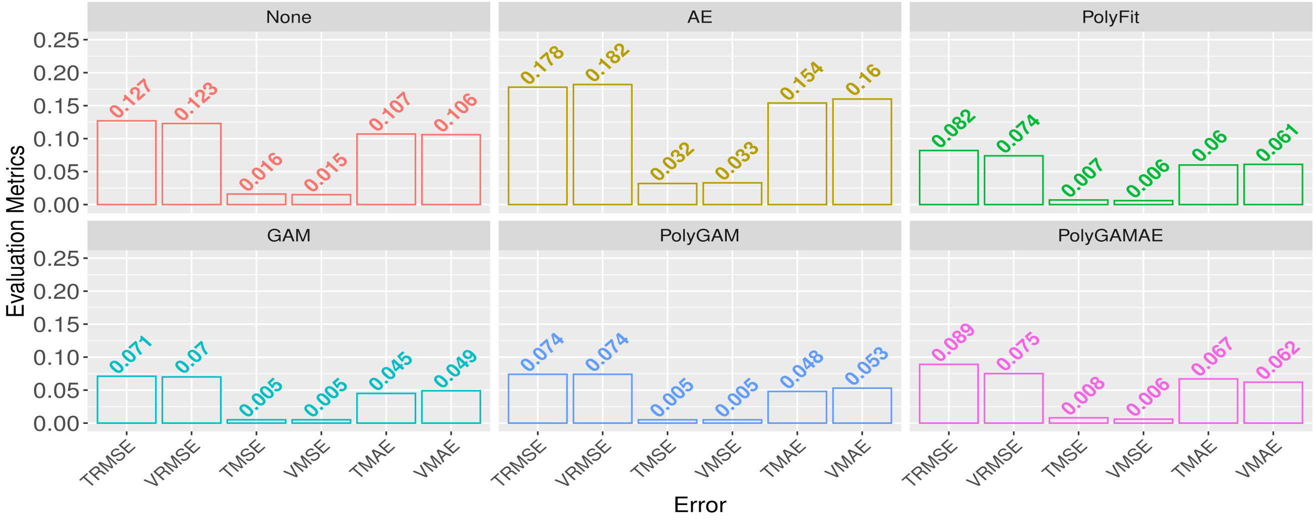


FIGURE 6. Variations in RMSE, MSE, and MAE for the training and validation sets when the ensemble learners use each of the six feature enhancement cases.

are labeled with a prefix "T") and validation phases (the errors are labeled with a prefix "V"). If the model is not overfitting or underfitting, the errors across training and validation sets should be comparable. This is true for all six feature enhancement cases, but comparing their magnitudes across the cases deduced that PolyGAM (in contention with GAM) has the lowest errors among them all.

Another way to ensure the model has fit best to the data and has the potential to generalize well is to plot the R^2 scores of training and validation against each other. Ideally, a best-fit model should have its scores align perfectly with the diagonal reference line. If the model fits better with validation set, the plotted values tend to be above the line. Figure 7 shows that except AE, the ensemble learners that use other feature enhancement cases either have similar scores (PolyGAM and GAM) or do better with the validation set (None, PolyFit, and PolyGAMAE). Although this is not indicative of the model's generalization performance, the figure suggests low overfit and underfit.

The Manhattan and Euclidean distances measured on normalized data are shown in Table 7 for AE (baseline),

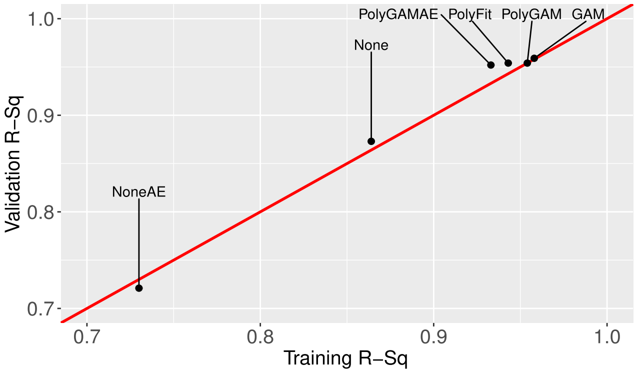
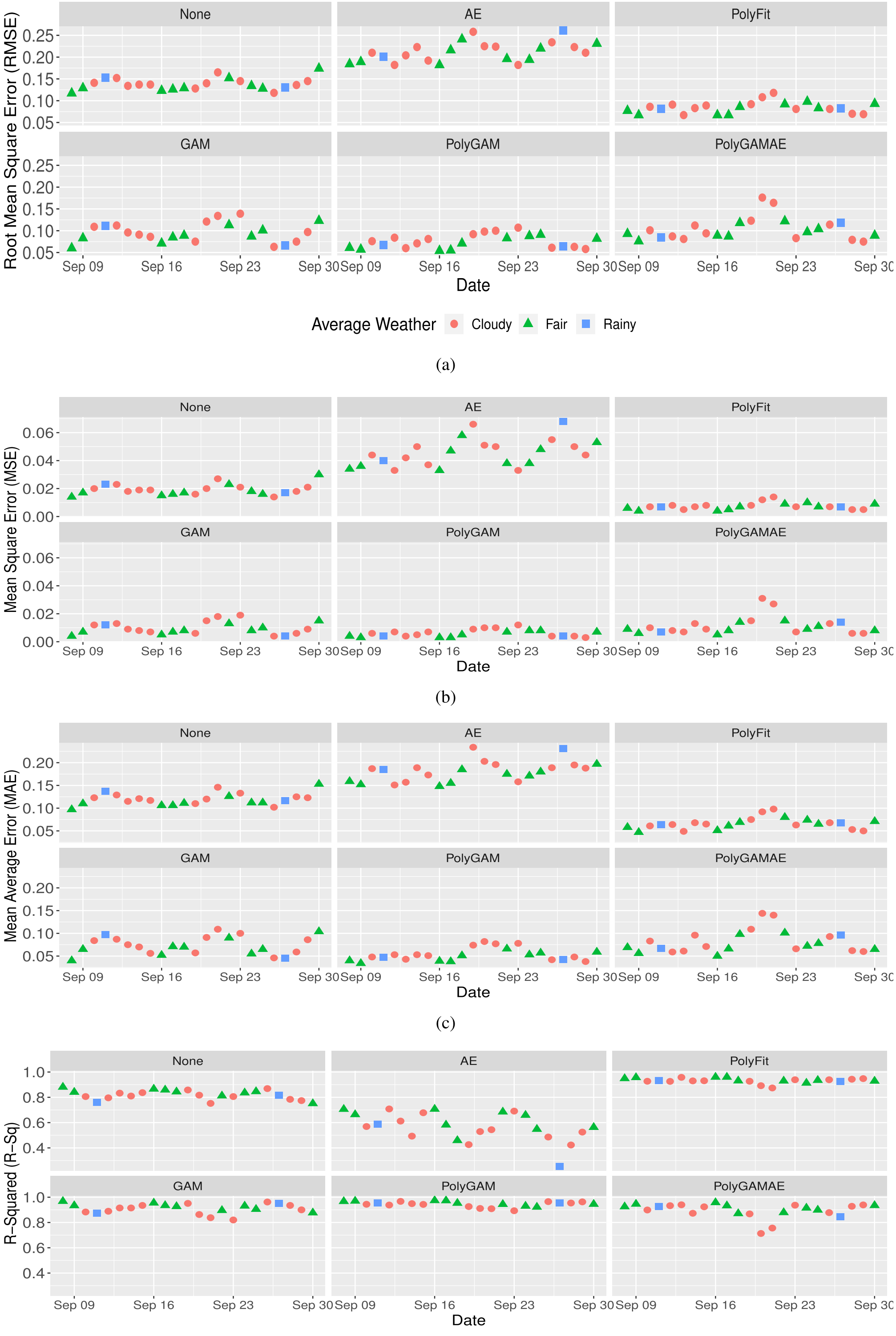


FIGURE 7. Plot of the R-squares between the training and validation sets.

PolyGAM (proposed approach), and PolyGAMAE (a hybrid of the two) cases. The model performances across each case seen earlier are reflected here too, with distance values being the least for PolyGAM on each data set. Furthermore, the distance values keep rising for the other two cases and have a range of 18 units for AE and 9 units for PolyGAMAE across different test day types. However, the distance values



(a)

(b)

(c)

(d)

FIGURE 8. Trends of (a) RMSE, (b) MSE, (c) MAE, and (d) R-square values for the ERM across test days (September 8–30, 2019) categorized into cloudy (red circle), fair/sunny (green triangle), and rainy (blue square) days. The trends denote a periodicity in performance irrespective of the feature enhancement strategy.

remain relatively consistent for PolyGAM with a range of 6 units across the test day types. This would not be the case if model drift occurred over time, in which the distances would show a significant, continuously rising trend.

V. DISCUSSION OF RESULTS

The consistency of PolyGAM as seen in the previous section suggests that the ERM, once trained using features enhanced by PolyGAM, can sustain a good order of predictions better

TABLE 6. A summary of errors for different data: training set, validation set, and three different testing sets (one for each weather: fair, cloudy, and rainy).

Set	Mode	RMSE	MSE	MAE	R-Sq.
Train (Sept. 1–4)	None	0.127	0.016	0.107	0.864
	AE	0.178	0.032	0.154	0.73
	PolyFit	0.082	0.007	0.06	0.943
	GAM	**0.071**	**0.005**	**0.045**	**0.958**
	PolyGAM	0.074	0.005	0.048	0.954
	PolyGAMAE	0.089	0.008	0.067	0.933
Validation (Sept. 5–7)	None	0.123	0.015	0.106	0.873
	AE	0.182	0.033	0.16	0.721
	PolyFit	0.074	0.006	0.061	0.954
	GAM	**0.07**	**0.004**	**0.04**	**0.959**
	PolyGAM	0.074	0.005	0.053	0.954
	PolyGAMAE	0.075	0.006	0.062	0.952
Test (Sept. 9) (Fair)	None	0.129	0.017	0.11	0.841
	AE	0.189	0.036	0.152	0.664
	PolyFit	0.067	0.004	0.047	0.957
	GAM	0.083	0.007	0.065	0.934
	PolyGAM	**0.057**	**0.003**	**0.034**	**0.97**
	PolyGAMAE	0.076	0.006	0.056	0.946
Test (Sept. 15) (Cloudy)	None	0.137	0.019	0.117	0.837
	NoneAE	0.192	0.037	0.173	0.678
	PolyFit	0.089	0.008	0.065	0.931
	GAM	0.086	0.007	0.056	0.935
	PolyGAM	**0.081**	**0.007**	**0.051**	**0.943**
	PolyGAMAE	0.094	0.009	0.071	0.924
Test (Sept. 27) (Rainy)	None	0.13	0.017	0.117	0.816
	AE	0.261	0.068	0.231	0.255
	PolyFit	0.082	0.007	0.068	0.926
	GAM	0.066	0.004	0.045	0.952
	PolyGAM	**0.064**	**0.004**	**0.042**	**0.956**
	PolyGAMAE	0.118	0.014	0.096	0.847

TABLE 7. Manhattan (M) and euclidean (E) distances to quantify model drift through data similarity evaluation.

Set	AE		PolyGAM		PolyGAMAE	
	M	E	M	E	M	E
Train	38, 247	89	16, 150	47	25, 536	62
Validation	31, 777	79	12, 047	34	13, 984	39
Test (Sept. 9) (Fair)	9, 907	50	4, 386	21	5, 574	26
Test (Sept. 15) (Cloudy)	11, 164	48	4, 903	27	5, 556	30
Test (Sept. 27) (Rainy)	15, 667	68	4, 846	23	7, 863	35

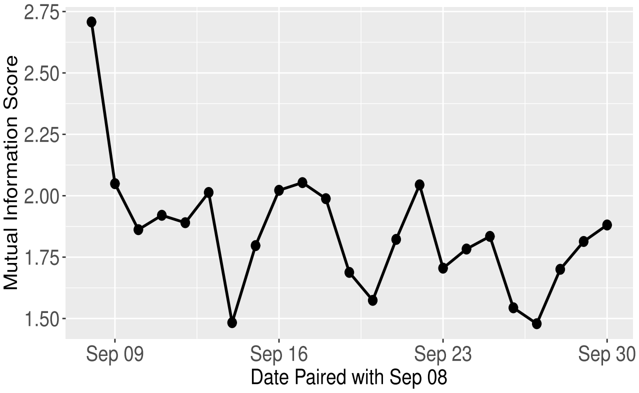


FIGURE 9. The variation in averaged mutual information scores between actual and predicted PV power across all test dates.

than when it does not use PolyGAM-enhanced features. This finding is further reinforced in Fig. 8 where the test error scores and R^2 values are shown across the test dates for each case. The type of weather prevalent in each day is also reflected, with solid green triangles for fair, solid red circles for cloudy, and solid blue squares for rainy conditions. The results presented in the figures and tables of Section IV are in agreement with Fig. 8. Looking at the variations in error value magnitudes, PolyGAM shows the least deviations, followed by PolyFit, GAM, None, PolyGAMAE, and finally AE. The error values are consistently higher for cloudy and rainy days compared with fair days. Likewise, the R^2 values dip when the weather is unfavorable. This demonstrates that although the ERM is resilient in handling patterns ahead in time, there is room for improvement. Further investigation will be conducted in the future.

Therefore, ERM can be leveraged to make retraining less frequent than if PolyGAM were not used. However, the resulting feature space FS_{PG} and the ERM must retain their generalization as the gap between the testing date and training-validation dates increases. The generalization of a model is measured in how effectively it adapts to previously unseen data. Because of the reasons outlined in Section I when introducing the concept of model drift, retraining becomes an essential part of the predictor life cycle.

Two metrics are leveraged to quantify drift, which is the sole criterion within the scope of this paper. Figure 9 depicts the mutual information score between actual and predicted PV power for each of the test dates, averaged across all observations for the day. A high score indicates a greater agreement between the observations. The score is highest for September 8, the date that follows the training-validation set dates. The score is the lowest for September 14, 20, and 27. The first two dates correspond to cloudy days and the third to a rainy day. The score is relatively low except for September 9, 13, 16–18, 22, and 30. All of these dates, except for September 13, experienced fair weather. Thus, the intricate relationship between mutual information scores and the day's weather is reflected in the FEEL framework performance. However, overall, the score remains consistent and does not taper close to 0, which would indicate a disagreement between observations.

Some of the most recent works in the literature that proposed and validated techniques similar in scope and contribution to this paper are used for one-on-one comparison of performance, the results of which are summarized in Table 8.

TABLE 8. Comparison of the performance of different methods proposed in the literature against the proposed FEEL framework.

Ref.	Method	Core Concept	RMSE		MSE		MAE	
			Ref.	FEEL	Ref.	FEEL	Ref.	FEEL

[21]	Adaptive hybrid model for day-ahead prediction of PV output power	An ensemble of improved variational mode decomposition, ARIMA, and improved deep belief network	0.075: F^{a};
			0.084: C;
			0.1: R
				0.057: F;
				0.079: C;
				0.065: R
					–	**0.005: F;**
						0.007: C;
						0.004: R
							0.048: F;
							0.064: C;
							0.073: R
								0.048: F;
								0.057: C;
								0.044: R

[22]	PV power prediction in a peak zone	An ensemble of LSTM and gate recurrent network	0.14 (next-day-ahead)	**0.061** (next-day-ahead)	–	0.004 (next-day-ahead)	**0.015** (next-day-ahead)	0.04 (next-day-ahead)
[23]	Day-ahead PV power forecasting under partial daily pattern prediction	An ensemble of LSTM and time correlation modification	**0.063** across all test days	0.074 across all test days	–	**0.006** across all test days	**0.028** across all test days	0.052 across all test days
[24]	Data-driven day-ahead PV estimation	An ensemble of autoencoder, LSTM, and persistence model	0.084: μ^{b}; 0.053: σ^{b}	**0.075**: μ; **0.015**: σ	–	0.006: μ; 0.003: σ	**0.046**: μ; 0.029: σ	0.053: μ; **0.014**: σ
[25]	Multiple-input deep CNN for short-term PV power forecasting	A deep CNN model trained on past 5 day-data for next-day-ahead forecast	1.632 (next-day-ahead)	**0.061** (next-day-ahead)	–	0.004 (next-day-ahead)	1.095 (next-day-ahead)	**0.04** (next-day-ahead)
[26]	Short-term PV power prediction	An ensemble of daily feature matrix and LSTM	0.296: sun^{c}; 1.407: mixed^{c}	**0.06**: sun; **0.069**: mixed	–	0.004: sun; 0.005: mixed	–	0.042: sun; 0.044: mixed
[27]	Day-ahead PV power forecasting	An ensemble of CNN and LSTM with enhanced attention and shortcut connection	0.111 (next-day-ahead)	**0.061** (next-day-ahead)	–	0.004 (next-day-ahead)	–	0.04 (next-day-ahead)

[a] F: Fair day, C: cloudy day, R: rainy day
[b] μ: mean error score, σ: standard deviation of error score
[c] sun: fair day, mixed: fair, cloudy, and rainy days

The FEEL framework exhibits comparable performance to each of these techniques. The error measures, redesigned in accordance with the metrics used by each of these works, show that the FEEL framework has lower RMSE scores than that of the reference frameworks in six of the seven evaluations. This is shown by the bold-faced text in the RMSE column. Differences in hardware specifications, validation system and parameter initializations, PV plant(s) considered, time of the year, and inherent model randomness must be acknowledged as factors that might reduce the interpretability of these results, but despite those factors, the performance of the FEEL framework reinforces $\mathcal{H}_B$, that the model can improve but also sustain the generalization of the PV power predictors.

VI. CONCLUSION AND FUTURE WORK

This paper formulated and implemented a FEEL framework that uses a combination of PolyFit and GAM to capture linear and nonlinear partial dependencies between the feature set $FS = [I, WS, AT, RH, AP]$ and the target variable Gen. The work also defined five other feature enhancement cases: no enhancement (None), AE (the baseline), only PolyFit, only GAM, and PolyGAMAE. The 6 cases were individually applied to a 10-model ensemble of regularized MLP models whose predictions of Gen were aggregated to yield the final predictions. By training and validating the FEEL framework on a training set (September 1–4, 2019) and a validation set (September 5–7, 2019) for each case, the models were tested for the rest of the month that include days with fair, cloudy, and rainy weather conditions. Results show that PolyGAM-enhanced features FS_{PG} consistently perform better on test days of all three weather types. Additionally, when PolyGAM was integrated with the baseline AE model, the collective performance dramatically improved across all test days. Thus, $\mathcal{H}_A$ cannot be rejected. By virtue of minimal concept drift between the testing set and the training-validation set demonstrated by the Minkowski distance values, and the reduced risk of overfitting and underfitting as illustrated in Section IV, $\mathcal{H}_B$ cannot be rejected.

The FEEL framework, deployed and field-validated on an Edge-compute node, will be significant for PV power prediction in high-penetration scenarios of both distribution and transmission grids. Towards achieving this goal, the work can be expanded in the future by the following means:

1) Instead of selecting hyperparameters through grid search, optimization techniques can be devised to make the selection more robust and dynamic.
2) Although the model formulation stages in Section III can be replicated, the most effective feature enhancement strategy might vary depending on the location and mission profile of the PV plant. Future work could apply the proposed strategy on data from PV plants

across different geographical locations over multiple seasons to determine latent patterns.

3) The proposed strategy can be extended by taking forecasts (from physics-based techniques).
4) The framework's explainability can be improved by augmenting probabilistic and quantile regression-based predictors alongside the ERMs.
5) Considering the emerging paradigm that will shift computational intelligence partially to the grid-edge, the FEEL framework can be migrated from a batch train-and-validate environment to a time-series sequence environment in which online learning can be integrated to make the framework more lightweight and robust.

ACKNOWLEDGMENT

This manuscript has been authored by UT-Battelle, LLC based upon work supported by the U.S. Department of Energy, Office of Electricity under contract number DE-AC05-00OR22725. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for U.S. Government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

REFERENCES

[1] A. I. Sarwat, M. Amini, A. Domijan, Jr., A. Damnjanovic, and F. Kaleem, "Weather-based interruption prediction in the smart grid utilizing chronological data," *J. Mod. Power Syst. Clean Energy*, vol. 4, no. 2, pp. 308–315, Apr. 2016.

[2] A. Anzalchi, A. Sundararajan, L. Wei, A. Moghadasi, and A. I. Sarwat, "Future directions to the application of distributed fog computing in smart grid systems," in *Smart Grid Analytics for Sustainability and Urbanization*. Hershey, PA, USA: IGI Global, 2018, pp. 162–195.

[3] A. Sundararajan and A. I. Sarwat, "Hybrid data-model method to improve generation estimation and performance assessment of grid-tied PV: A case study," *IET Renew. Power Gener.*, vol. 13, no. 13, pp. 2480–2490, Oct. 2019.

[4] I. Colak, M. Yesilbudak, N. Genc, and R. Bayindir, "Multi-period prediction of solar radiation using ARMA and ARIMA models," in *Proc. IEEE 14th Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2015, pp. 1045–1049.

[5] S. I. Vagropoulos, G. I. Chouliaras, E. G. Kardakos, C. K. Simoglou, and A. G. Bakirtzis, "Comparison of SARIMAX, SARIMA, modified SARIMA and ANN-based models for short-term PV generation forecasting," in *Proc. IEEE Int. Energy Conf. (ENERGYCON)*, Apr. 2016, pp. 1–6.

[6] S. Hussain and A. A. Alili, "Day ahead hourly forecast of solar irradiance for Abu Dhabi, UAE," in *Proc. IEEE Smart Energy Grid Eng. (SEGE)*, Aug. 2016, pp. 68–71.

[7] M. Z. Hassan, M. E. K. Ali, A. B. M. S. Ali, and J. Kumar, "Forecasting day-ahead solar radiation using machine learning approach," in *Proc. 4th Asia–Pacific World Congr. Comput. Sci. Eng. (APWC CSE)*, Dec. 2017, pp. 252–258.

[8] G. Vanderstar, P. Musilek, and A. Nassif, "Solar forecasting using remote solar monitoring stations and artificial neural networks," in *Proc. IEEE Can. Conf. Electr. Comput. Eng. (CCECE)*, May 2018, pp. 1–4.

[9] K. Chen, Z. He, K. Chen, J. Hu, and J. He, "Solar energy forecasting with numerical weather predictions on a grid and convolutional networks," in *Proc. IEEE Conf. Energy Internet Energy Syst. Integr. (EI2)*, Nov. 2017, pp. 1–5.

[10] M. Abuella and B. Chowdhury, "Solar power probabilistic forecasting by using multiple linear regression analysis," in *Proc. SoutheastCon*, Apr. 2015, pp. 1–5.

[11] F. Serttas, F. O. Hocaoglu, and E. Akarslan, "Short term solar power generation forecasting: A novel approach," in *Proc. Int. Conf. Photovoltaic Sci. Technol. (PVCon)*, Jul. 2018, pp. 1–4.

[12] J. Xu, S. Yoo, D. Yu, D. Huang, J. Heiser, and P. Kalb, "Solar irradiance forecasting using multi-layer cloud tracking and numerical weather prediction," in *Proc. 30th Annu. ACM Symp. Appl. Comput.*, New York, NY, USA, Apr. 2015, pp. 2225–2230, doi: 10.1145/2695664.2695812.

[13] M. Chai, F. Xia, S. Hao, D. Peng, C. Cui, and W. Liu, "PV power prediction based on LSTM with adaptive hyperparameter adjustment," *IEEE Access*, vol. 7, pp. 115473–115486, 2019.

[14] S.-G. Kim, J.-Y. Jung, and M. Sim, "A two-step approach to solar power generation prediction based on weather data using machine learning," *Sustainability*, vol. 11, no. 5, p. 1501, Mar. 2019.

[15] G. G. Kim, J. H. Choi, S. Y. Park, B. G. Bhang, W. J. Nam, H. L. Cha, N. Park, and H.-K. Ahn, "Prediction model for PV performance with correlation analysis of environmental variables," *IEEE J. Photovolt.*, vol. 9, no. 3, pp. 832–841, May 2019.

[16] H. Eom, Y. Son, and S. Choi, "Feature-selective ensemble learning-based long-term regional PV generation forecasting," *IEEE Access*, vol. 8, pp. 54620–54630, 2020.

[17] B. Chen, P. Lin, Y. Lai, S. Cheng, Z. Chen, and L. Wu, "Very-short-term power prediction for PV power plants using a simple and effective RCC-LSTM model based on short term multivariate historical datasets," *Electronics*, vol. 9, no. 2, p. 289, Feb. 2020.

[18] M. Afrasiabi, M. Mohammadi, M. Rastegar, and S. Afrasiabi, "Deep learning architecture for direct probability density prediction of small-scale solar generation," *IET Gener., Transmiss. Distrib.*, vol. 14, no. 11, pp. 2017–2025, 2020.

[19] C. J. Huang, Y. Shen, Y. H. Chen, and H. C. Chen, "A novel hybrid deep neural network model for short-term electricity price forecasting," *Int. J. Energy Res.*, vol. 45, no. 2, pp. 2511–2532, 2020.

[20] D. S. Tripathy, B. R. Prusty, and K. Bingi, "Reliable prediction intervals of PV generation using quantile regression averaging approach," in *Proc. 3rd Int. Conf. Energy, Power Environ.: Towards Clean Energy Technol.*, Mar. 2021, pp. 1–6.

[21] J. Zhang, Z. Tan, and Y. Wei, "An adaptive hybrid model for day-ahead photovoltaic output power prediction," *J. Cleaner Prod.*, vol. 244, Jan. 2020, Art. no. 118858.

[22] D. Lee and K. Kim, "PV power prediction in a peak zone using recurrent neural networks in the absence of future meteorological information," *Renew. Energy*, vol. 173, pp. 1098–1110, Aug. 2021.

[23] F. Wang, Z. Xuan, Z. Zhen, K. Li, T. Wang, and M. Shi, "A day-ahead PV power forecasting method based on LSTM-RNN model and time correlation modification under partial daily pattern prediction framework," *Energy Convers. Manage.*, vol. 212, May 2020, Art. no. 112766.

[24] Y. Zhang, C. Qin, A. K. Srivastava, C. Jin, and R. Sharma, "Data-driven day-ahead PV estimation using autoencoder-LSTM and persistence model," *IEEE Trans. Ind. Appl.*, vol. 56, no. 6, pp. 7185–7192, Nov./Dec. 2020.

[25] C.-J. Huang and P.-H. Kuo, "Multiple-input deep convolutional neural network model for short-term photovoltaic power forecasting," *IEEE Access*, vol. 7, pp. 74822–74834, 2019.

[26] R. Zheng, G. Li, K. Wang, B. Han, Z. Chen, and M. Li, "Short-term photovoltaic power prediction based on daily feature matrix and deep neural network," in *Proc. 6th Asia Conf. Power Electr. Eng. (ACPEE)*, Apr. 2021, pp. 290–294.

[27] Y. Wang, Y. Chen, H. Liu, X. Ma, X. Su, and Q. Liu, "Day-ahead photovoltaic power forcasting using convolutional-LSTM networks," in *Proc. 3rd Asia Energy Electr. Eng. Symp. (AEEES)*, Mar. 2021, pp. 917–921.

[28] M. Frasco, "Evaluation metrics for machine learning," in *R Project Documentation Metrics Package*. 2018, pp. 1–26. [Online]. Available: https://CRAN.R-project.org/package=Metrics

[29] J. Walters-Williams and Y. Li, "Estimation of mutual information: A survey," in *Rough Sets and Knowledge Technology*, P. Wen, Y. Li, L. Polkowski, Y. Yao, S. Tsumoto, and G. Wang, Eds. Berlin, Germany: Springer, 2009, pp. 389–396.

[30] H. Liu, J. Sun, L. Liu, and H. Zhang, "Feature selection with dynamic mutual information," *Pattern Recognit.*, vol. 42, no. 7, pp. 1330–1339, Jul. 2009.

[31] I. Batyrshin, "Constructing time series shape association measures: Minkowski distance and data standardization," in *Proc. BRICS Congr. Comput. Intell. 11th Brazilian Congr. Comput. Intell.*, Sep. 2013, pp. 204–212.

[32] O. Ajala, C. Jackson, J. Thompson, A. Garg, and B. Ollis, "Evaluating the technical feasibility of microgrids provide local neighborhood support, reliability, and other grid benefits," Oak Ridge Nat. Lab., Joint Rep. Southern Company Services, Electr. Power Res. Inst., Oak Ridge, TN, USA, Tech. Rep., 2019.

[33] A. Sundararajan and A. I. Sarwat, "Roadmap to prepare distribution grid-tied photovoltaic site data for performance monitoring," in *Proc. Int. Conf. Big Data, IoT Data Sci. (BID)*, Dec. 2017, pp. 110–115.

[34] A. Sundararajan, T. O. Olowu, L. Wei, S. Rahman, and A. I. Sarwat, "Case study on the effects of partial solar eclipse on distributed PV systems and management areas," *IET Smart Grid*, vol. 2, no. 3, pp. 477–490, Jun. 2019.

[35] A. I. Sarwat and A. Sundararajan, "Distributed renewable energy grid controller," U.S. Patent 10 326 280, Jun. 18, 2019. [Online]. Available: http://www.freepatentsonline.com/10326280.html

[36] A. Anzalchi, A. Sundararajan, A. Moghadasi, and A. Sarwat, "High-penetration grid-tied photovoltaics: Analysis of power quality and feeder voltage profile," *IEEE Ind. Appl. Mag.*, vol. 25, no. 5, pp. 83–94, Sep. 2019.

[37] A. Sundararajan, T. Khan, H. Aburub, A. I. Sarwat, and S. Rahman, "A tri-modular Human-on-the-Loop framework for intelligent smart grid cyber-attack visualization," in *Proc. SoutheastCon*, Apr. 2018, pp. 1–8.

[38] A. Sundararajan and A. I. Sarwat, "Evaluation of missing data imputation methods for an enhanced distributed PV generation prediction," in *Advances in Intelligent Systems and Computing*. Cham, Switzerland: Springer, Oct. 2019, pp. 590–609.

[39] O. Ceylan, M. Starke, P. Irminger, B. Ollis, D. King, and K. Tomsovic, "Hourly day ahead solar irradiance forecasting model in LabVIEW using cloud cover data," *Electr. J.*, vol. 16, no. 2, pp. 2047–2054, 2016.

[40] O. Ceylan, M. Starke, P. Irminger, B. Ollis, and K. Tomsovic, "A regression based hourly day ahead solar irradiance forecasting model by labview using cloud cover data," in *Proc. 9th Int. Conf. Electr. Electron. Eng. (ELECO)*, Nov. 2015, pp. 406–410.

[41] A. McIlvenna, A. Herron, J. Hambrick, B. Ollis, and J. Ostrowski, "Reducing the computational burden of a microgrid energy management system," *Comput. Ind. Eng.*, vol. 143, May 2020, Art. no. 106384.

[42] T. Yang, B. Li, and Q. Xun, "LSTM-attention-embedding model-based day-ahead prediction of photovoltaic power output using Bayesian optimization," *IEEE Access*, vol. 7, pp. 171471–171484, 2019.

[43] H.-C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, and M. O. Leach, "Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1930–1943, Aug. 2013.

[44] A. Gensler, J. Henze, B. Sick, and N. Raabe, "Deep Learning for solar power forecasting—An approach using autoencoder and LSTM neural networks," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2016, pp. 002858–002865.

[45] S. Tasnim, A. Rahman, A. M. T. Oo, and M. E. Hague, "Autoencoder for wind power prediction," *Renewables*, vol. 4, no. 6, pp. 1–11, Dec. 2017.

[46] J. Mathe, N. Miolane, N. Sebastien, and J. Lequeux, "PVNet: A LRCN architecture for spatio-temporal photovoltaic PowerForecasting from numerical weather prediction," 2019, *arXiv:1902.01453*. [Online]. Available: http://arxiv.org/abs/1902.01453

[47] M. AlKandari and I. Ahmad, "Solar power generation forecasting using ensemble approach based on deep learning and statistical methods," *J. Appl. Comput. Inform.*, vol. 16, nos. 1–2, pp. 1–20, Aug. 2020.

[48] M. Lotfi, M. Javadi, G. J. Osório, C. Monteiro, and J. P. S. Catalão, "A novel ensemble algorithm for solar power forecasting based on kernel density estimation," *Energies*, vol. 13, no. 1, p. 216, Jan. 2020.

[49] L. Gigoni, A. Betti, E. Crisostomi, A. Franco, M. Tucci, F. Bizzarri, and D. Mucci, "Day-ahead hourly forecasting of power generation from photovoltaic plants," 2019, *arXiv:1903.06800*. [Online]. Available: http://arxiv.org/abs/1903.06800

[50] H. Nazaripouya, B. Wang, Y. Wang, P. Chu, H. R. Pota, and R. Gadh, "Univariate time series prediction of solar power using a hybrid wavelet-ARMA-NARX prediction method," in *Proc. IEEE/PES Transmiss. Distribution Conf. Expo. (T&D)*, May 2016, pp. 1–5.

[51] P.-H. Chiang, S. P. V. Chiluvuri, S. Dey, and T. Q. Nguyen, "Forecasting of solar photovoltaic system power generation using wavelet decomposition and bias-compensated random forest," in *Proc. 9th Annu. IEEE Green Technol. Conf. (GreenTech)*, Mar. 2017, pp. 260–266.

[52] D. Zhang, Y. Yu, and Z. Huang, "Forecasting solar power using wavelet transform framework based on ELM," in *Proc. ELM* (Proceedings in Adaptation, Learning and Optimization), vol. 10, J. Cao, C. Vong, Y. Miche, and A. Lendasse, Eds. Springer, Cham, 2019.

[53] P. Mandal, S. T. S. Madhira, A. U. Haque, J. Meng, and R. L. Pineda, "Forecasting power output of solar photovoltaic system using wavelet transform and artificial intelligence techniques," *Procedia Comput. Sci.*, vol. 12, pp. 332–337, 2012.

[54] W. Ma, X. Sun, Y. Song, F. Tao, W. Feng, Y. He, N. Zhao, and Z. Yuan, "Applied mixed generalized additive model to assess the effect of temperature on the incidence of bacillary dysentery and its forecast," *PLoS ONE*, vol. 8, no. 4, Apr. 2013, Art. no. e62122.

[55] M. Brabec, M. Paulescu, and V. Badescu, "Generalized additive models for nowcasting cloud shading," *Sol. Energy*, vol. 101, pp. 272–282, Mar. 2014.

[56] Y. Goude, R. Nedellec, and N. Kong, "Local short and middle term electricity load forecasting with semi-parametric additive models," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 440–446, Jan. 2014.

[57] C. Molnar, "A guide for making black box models explainable: Partial Dependence plot (PDP)," in *Interpretable Machine Learning*. Apr. 2020, Ch. 5. [Online]. Available: https://christophm.github.io/interpretable-ml-book/

[58] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2010, pp. 1–8.

[59] B. L. Kalman and S. C. Kwasny, "Why tanh: Choosing a sigmoidal function," in *Proc. Int. Joint Conf. Neural Netw.*, vol. 4, Jun. 1992, pp. 578–581.

[60] B. Karlik and A. V. Olgac, "Performance analysis of various activation functions in generalized MLP architectures of neural networks," *Int. J. Artif. Intell. Expert Syst. (IJAE)*, vol. 1, no. 4, pp. 111–122, 2011.

[61] H. Zheng, Z. Yang, W. Liu, J. Liang, and Y. Li, "Improving deep neural networks using softplus units," in *Proc. IJCNN*, Jul. 2015, pp. 1–4.

[62] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2015, pp. 1–15.

[63] J. L. Rodgers and W. A. Nicewander, "Thirteen ways to look at the correlation coefficient," *Amer. Statistician*, vol. 42, no. 1, pp. 52–66, 1988.

[64] F. A. Farris, "The Gini index and measures of inequality," *Amer. Math. Monthly*, vol. 117, no. 10, pp. 851–864, 2017.

[65] S. Yitzhaki, "The GMD: A superior measure of variability for non-normal distributions," *Metron*, vol. 61, no. 2, pp. 285–316, Feb. 2003. [Online]. Available: https://ssrn.com/abstract=301740

[66] R. La Haye and P. Zizler, "The Gini mean difference and variance," *METRON*, vol. 77, no. 1, pp. 43–52, Apr. 2019.

[67] C. Xiao, J. Ye, R. M. Esteves, and C. Rong, "Using Spearman's correlation coefficients for exploratory data analysis on big dataset," *Concurrency Comput., Pract. Exper.*, vol. 28, no. 14, pp. 3866–3878, Sep. 2016.

ADITYA SUNDARARAJAN (Member, IEEE) received the M.S. degree in computer engineering and the Ph.D. degree in electrical and computer engineering from Florida International University, Miami, in 2014 and 2019, respectively. Since 2020, he has been a Research and Development Associate Staff Member with Oak Ridge National Laboratory, Grid Components and Controls Group. He has more than 50 publications in renowned journals and conference proceedings. His research interests include applied machine learning, artificial intelligence, and data analytics to address challenges in renewable integration, intelligent grids, systems development, and situation awareness. He received a best paper award, in 2017.

BEN OLLIS (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from The University of Tennessee, Knoxville, in 2012 and 2014, respectively. He has been a member of Oak Ridge National Laboratory's Grid Components and Controls Group, since 2013. He is a Research and Development Staff Member and a Power Systems Research Engineer at Oak Ridge National Laboratory. He has utility experience as a Planning Engineer at Duke Energy and as a Power System Operator with Clinton Utilities Board. He currently leads research projects focused on small-scale and large-scale microgrids, grid resiliency, device interoperability, and novel energy generation technologies.