Dynamic Feature Selection for Solar Irradiance Forecasting Based on Deep Reinforcement Learning

Cheng Lyu, *Graduate Student Member, IEEE*, Sara Eftekharnejad, *Senior Member, IEEE*, Sagnik Basumallik, *Member, IEEE*, and Chongfang Xu

***Abstract*—A large volume of data is typically needed to achieve an accurate solar generation prediction. However, not all types of data are consistently available. Various research efforts have addressed this challenge by developing methods that identify the most relevant features for predicting solar generation. However, the optimal features vary with different weather patterns, making it impossible to select a fixed set of optimal features for all weather patterns. This study develops a new framework to accurately predict solar irradiance using dynamically changing optimal features. The developed model first incorporates feature extraction with clustering techniques to identify representative weather data from a dataset. Next, using deep reinforcement learning (DRL), a new feature selection method is developed to yield the minimum features required to accurately forecast solar irradiance from representative data. Benefiting from the model-free nature of DRL, the developed method is adaptive to various weather conditions, and dynamically alters the selected features. Case studies using real-world data have shown that the developed model significantly reduces the volume of data required for accurate irradiance forecasting for different weather patterns.**

***Index Terms*—Data analytics, data clustering, deep reinforcement learning, feature extraction, solar generation forecast.**

NOMENCLATURE

$Q(s_t, a_t)$	The Q-value for the state s_t and the action a_t
α	Eigenvectors of centered Kernel Matrix in KPCA.
ϵ, μ	Parameters controlling penalty for T and w
γ, β	Reward decay factor, and learning rate in DQN.
$\hat{y}$	Predicted results of XGBoost.
λ_i, A_i	ith eigenvalue, and ith eigenvector.
$\mathcal{I}, \mathcal{I}_L, \mathcal{I}_R$	All nodes, left nodes, and right nodes in a tree.
$\Omega(f_k)$	Complexity penalty of kth tree in XGBoost.
$\Phi(X)$	A higher dimensional dataset transformed from the original weather dataset.
σ	Standard deviation.
θ_i	Inner parameters (weights and bias) of the present Q-network at the ith iteration.
θ_i^-	Inner parameters (weights and bias) for target Q-network at the ith iteration.
a	A weighting constant for reward.
AF	Number of the features used for forecast.
b_i	Size of selected data from each cluster.
c_i	Centroid for the ith cluster.
f_k	kth independent regression tree with structure q and leaf weights w
$l()$	Loss function in XGBoost.
$Loss_i(\theta)$	The value of the loss function at the ith iteration given parameter θ in DQN.
m, t	Dimension of original weather dataset X, and $\Phi(X)$
N	Number of data points in the original weather dataset.
n_i	Number of data points in the ith cluster.
n_{si}	The selected representative data for the ith cluster.
Q^{old}, Q^{new}	Q-value of last iteration, and the updated Q-value.
s_t, a_t, r_t	State, action, and reward at step t, respectively.
T, w	Number of leafs, and leaf weights in a tree.
W, D, L	Similarity matrix, Diagonal matrix, and Laplacian matrix, respectively.
$X(x)$	The original weather dataset.
$x_i^{(j)}$	The ith data in the jth cluster.
X_{repre}	The representative dataset extracted from the original weather dataset.
X_{trans}	The output of KPCA and the input to SC.
y	Label when training XGBoost.
Y_{actual}	Real solar irradiance.
Y_{pred}	Predicted solar irradiance with the selected features.

Manuscript received 13 January 2022; revised 20 April 2022 and 14 July 2022; accepted 27 August 2022. Date of publication 15 September 2022; date of current version 19 January 2023. Paper 2021-IACC-1658.R2, presented at the 2021 IEEE Texas Power and Energy Conference, College Station, TX, USA, Feb. 02-05, and approved for publication in the IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS by the Industrial Automation and Control Committee of the IEEE Industry Applications Society. This work was supported by the National Science Foundation under Grant 2144918. *(Corresponding author: Sara Eftekharnejad.)*

Cheng Lyu and Sara Eftekharnejad are with the Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY 13244 USA (e-mail: clyu07@syr.edu; sara.eftekharnejad@ieee.org).

Sagnik Basumallik is with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV 13244 USA (e-mail: sagnik.basumallik@mail.wvu.edu).

Chongfang Xu is with the Department of Electrical and Computer Engineering, University of Washington, Seattle, WA 98195 USA (e-mail: cxu105@syr.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIA.2022.3206731.

Digital Object Identifier 10.1109/TIA.2022.3206731

I. INTRODUCTION

WITH the unprecedented growth of solar-powered generation, an accurate forecast of solar generation has become critical to grid operations, notably for maintaining load and generation balance in real time. Traditional solar forecasting

methods rely on the availability of large-scale historical weather data (e.g., temperature, humidity, and wind speed) and often suffer from the following shortcomings:

1) They require large-scale historical data to guarantee forecast accuracy while accessing massive weather data is limited for many utilities;
2) There is a need to balance data adequacy with avoiding over-fitting problems.

Due to effectively reducing the volume of data used, feature selection is proven to be an effective solution to these limitations [1]. However, the optimal features fluctuate depending on weather conditions; thus, choosing a single set of fixed features under all circumstances will result in a sub-optimal solution. Consequently, it is essential to develop a framework for predicting solar generation that is adaptable to changing weather conditions. This study aims to build a dynamic model for solar forecasting based on Deep Reinforcement Learning (DRL). The developed model yields optimal features for day-ahead or hour-ahead solar irradiance forecasting that are adapted to different weather patterns.

To demonstrate the new ideas pursued in this paper, we will first discuss the existing applications of Reinforcement Learning (RL) in generation forecasting. Current applications can be grouped into three categories: 1) optimizing the parameters of existing forecasting models [2], [3]; 2) optimizing time-series forecasting results [4]; 3) real-time selection of optimal predictive models [5], [6]. The authors in [2], [3] optimized the initial parameters of an RL-based neural network (NN) predictive model to avoid local minima when training a NN model. A hybrid model for short-term wind speed forecasting is developed in [4], which combines Empirical Wavelet Transform (EWT), Deep Network, and RL for improved forecast results. Upon decomposing the original data into different sub-data, RL determines the best forecasting result among the results from each sub-data. However, this approach necessitates a significant amount of computing power, whereas more straightforward probabilistic methods, such as the Bayesian approach, can achieve similar goals. Alternatively, in [5], [6] the authors proposed using several forecast models, each generating a different forecast result. An RL agent determines the best model depending on real-time conditions. In the aforementioned studies, RL has been utilized solely to improve the accuracy of the existing forecast models. This improvement comes at the cost of increased computational expense, only to result in a slight improvement in accuracy. This paper develops a fundamentally different approach to utilizing RL to reduce the computational burden of data-driven forecast models.

RL has proven to be an effective tool for feature selection [7], [8], [9]. However, its performance is limited in higher dimensions, such as those anticipated in solar irradiance prediction. Combining RL and Deep Learning for advanced DRL-based feature selection provides the benefit of reliably capturing minimal discriminating features for solar irradiance forecasting, reducing computational cost, and improving forecast performance. The deployment of DRL also allows dynamic feature selection, which implies that the optimal features change according to the conditions at the time of the forecast. As indicated in Fig. 1, the optimal dynamic features are used to forecast solar irradiance.

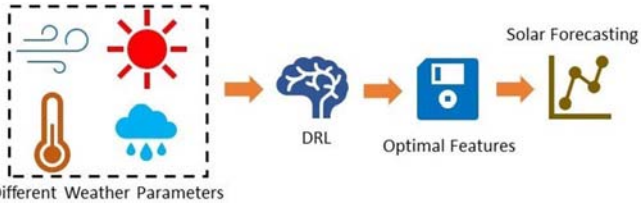


Fig. 1. Structure of the developed DRL-based dynamic feature selection method for solar irradiance forecasting.

The case studies presented in this paper have shown that the optimal features for forecasting solar irradiance during rainy conditions differ from those for sunny days. These results motivate the objective sought in this paper, which is to identify the minimum number of features for different weather patterns, and explore the feasibility of extending DRL for this purpose.

The main contributions of this paper can be listed as follows:

1) Developing a novel DRL-based method to identify a reduced set of optimal data for the prediction of solar irradiance. The optimal data used for prediction change dynamically, based on the local weather conditions;
2) The developed model can dramatically reduce the required data volume for an accurate forecast. Unlike traditional Artificial Neural Network (ANN)-based forecast models, the method presented in this paper generates similar forecast accuracy with much fewer data;
3) Case studies using real-world data demonstrate that the developed feature selection methodology strongly adapts to various weather conditions.

The rest of this paper is organized as follows. The state-of-the-art methods in solar generation forecasting are discussed in Section II. Details of the developed DRL-based model for solar irradiance forecasting are discussed in Section III. Case studies are presented in Section IV, followed by conclusions and future research in Section V.

II. State-of-The-Art Methods in Solar Generation Forecasting

Various recent efforts have aimed to forecast solar irradiance, which can be grouped into two main categories [10]: physics-based and Machine Learning (ML)-based methods. Physics-based models are classified into three main categories: models based on cloud imagery [11], models based on satellite data [12], and numerical weather prediction (NWP) models [13]. Nevertheless, the forecasting effectiveness of the physics-based models is limited by the volume of accessible data, i.e., the forecast efficiency is deficient since a considerable amount of raw data is required.

ML-based approaches have recently become common to estimate solar irradiance with large-scale historical data [14]. Among the existing methods, supervised learning, unsupervised learning, and RL are three main categories of interest in the research community. In terms of the supervised learning methods, Support Vector Machine (SVM), Regression, and Hidden Markov Model (HMM) are compared by the authors in [15] with diverse datasets from different locations for short-term solar irradiance forecasting. Similar comparisons are carried out in [16]

to compare Linear Regression, K-Nearest Neighbors (KNN), and SVM, while the SVM forecast model is shown to produce the most accurate forecast. Another comparative examination of several techniques, including Feedforward Neural Network (FFNN), Auto Regression (AR), KNN, and Markov Chain, is presented in [17], with FFNN demonstrated to be the optimal approach for solar irradiance forecasting among the four algorithms investigated. Although the aforementioned predictive models outperform traditional statistical methods, such as NWP, in terms of forecast accuracy, the models perform poorly during dynamic conditions, such as overcast days.

To further improve the forecast accuracy under dynamic conditions, hybrid models have been developed that combine multiple ML methods. In one study [15], Principal Component Analysis (PCA), primarily used for feature extraction, is coupled with Artificial Neural Network (ANN) for long-term solar irradiance forecasting. In another work [18], a hybrid solar forecasting model is established using sky images. Sky image data are first clustered with convolutional autoencoder and K-means. Upon grouping the original data into multiple clusters, several NN-based algorithms are leveraged to forecast solar power using distinct clusters. Combining the forecast results from each cluster results in the final forecast. Authors in [19] proposed a hybrid deep learning forecasting model that integrates auto-encoder long-short-term memory networks (AE-LSTM) with persistence model. AE-LSTM model is used to generate forecasts under complex weather conditions while the persistence model is only applied for continuous sunny weather conditions. Frequency-domain decomposition and deep learning are utilized in [20] for ultra short solar power forecasting. PV power is first decomposed into the low-frequency and high-frequency components. Convolutional neural networks (CNN) are then introduced to forecast both of the two components, from which the final forecasting result can be integrated to obtain. CNN is also integrated with satellite visible images processing to forecast minutely solar irradiance in [21]. Cloud cover is extracted from satellite visible images using a CNN. The generated cloud cover as well as other meteorological information, such as zenith angel, are used by a Multi-layer Perceptron (MLP) to yield forecasts.

To ensure the forecast accuracy of the aforementioned hybrid models, large-scale data are required. Such a volume of data may not always be available due to the need for continuous sensor measurement and storage. Therefore, identifying the optimal data for predicting solar irradiance can be critical to resolving the aforementioned issues while ensuring a minimal computational burden in both training and forecasting processes.

III. A New Hybrid Forecasting Model for Solar Irradiance Forecasting

In this paper, a fundamentally new approach to generation forecasting is proposed. DRL is extended to dynamically select the optimal features used for solar irradiance forecasting under various weather circumstances. The goal is to minimize the number of features used in the forecast horizon (e.g., hour-ahead) and reduce the computational burden, while ensuring an accurate forecast. Reduced computational burden generally leads to higher efficiency and faster computation speeds. A detailed description of the developed model is provided next.

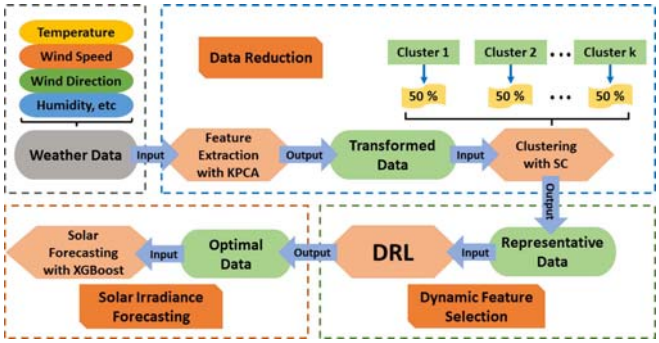


Fig. 2. Computing framework of the proposed forecasting model

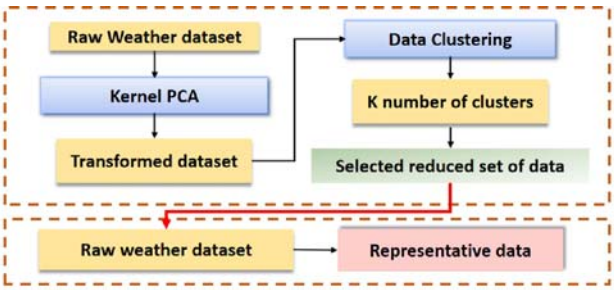


Fig. 3. The framework for the Data Reduction

A. An Overview of the Hybrid Forecasting Model

The enhanced solar irradiance forecast framework, illustrated in Fig. 2, incorporates feature extraction, data clustering, DRL, and the Extreme Gradient Tree Boosting (XGBoost) algorithms to produce an efficient and accurate forecast. Three major components work to attain this goal: data reduction, dynamic feature selection, and solar irradiance forecasting. The data reduction step reduces the scale of the raw datasets by clustering the data, while dynamic feature selection further reduces the dimension of data by finding the optimal features used for forecast.

The data reduction, illustrated in Fig. 3, incorporates feature extraction with data clustering and seeks to select representative data that best describe the original dataset. The dynamic feature selection process learns from the representative data to determine the optimal features using DRL. The developed XGBoost eventually uses the optimal features to yield solar irradiance forecasts.

Combining feature extraction with clustering is a practical technique for identifying representative data, as demonstrated by prior research [22], [23], [24]. This paper aims to dynamically reduce the size and dimension of the data required for an accurate prediction, significantly extending previous research [22]. In addition, the impact of diverse weather conditions on the size of the needed data for an accurate forecast has been thoroughly investigated. The comprehensive forecast model developed in this work dynamically selects a subset of data based on weather variations, which is the first of its kind approach in solar forecasting and offers a new angle for solar generation forecasting.

B. Data Reduction - Feature Extraction and Clustering

1) Feature Selection: Feature extraction is used to represent a large dataset with representative features [25]. Previous research works have shown that solar forecast accuracy can be enhanced by proper feature extraction [15], [22]. Linearly dependent data can be efficiently transformed using PCA, a common feature extraction methodology. However, the real-world data utilized for solar generation forecasts are nonlinearly dependent, and PCA is not applicable here. As an extension to PCA, Kernel Principal Component Analysis (KPCA) can be used for nonlinear data by incorporating a *kernel*. With specific kernel functions, KPCA enables the mapping of nonlinear data to a dot product feature space $\mathcal{F}$, where the new dataset Φ becomes linearly dependent [26]. Thus, the PCA component in KPCA can be applied to the new dataset Φ. This paper selects the Radial Basis Function (RBF) kernel function in equation (1) for mapping. As compared to other kernel functions, such as polynomial functions or sigmoid functions, RBF enhances the degree to which the output data become linearly separable when applied to the weather data used for solar forecasting [22].

$$K(x_i, x_j) = exp\left(-\frac{|x_i - x_j|^2}{2\sigma^2}\right) \tag{1}$$

The original weather data $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots \boldsymbol{x_N}\}$, $\boldsymbol{X} \in \mathbb{R}^{N\times m}$, where N is the number of data in $\boldsymbol{X}$ and m is the dimension of $\boldsymbol{X}$, is mapped to a higher dimensional space as:

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = (\boldsymbol{\Phi(x_i)} \cdot \boldsymbol{\Phi(x_j)}) = \boldsymbol{\Phi(x_i)\Phi(x_j)}^T, \quad i, j = 1, 2, \ldots, N \tag{2}$$

An N by N Kernel Matrix is thus constructed:

$$\boldsymbol{K} = \boldsymbol{\Phi(X)\Phi(X)^T} \tag{3}$$

where $\boldsymbol{K}_{ij} = K(x_i, x_j)$ and $\boldsymbol{\Phi(X)} = \{\boldsymbol{\Phi(x_1)}, \ldots, \boldsymbol{\Phi(x_N)}\}$. A higher dimensional dataset $\boldsymbol{\Phi(X)} \in \mathbb{R}^{N\times t}$, $(t > m)$, is defined in the dot product feature space $\mathcal{F}$. Performing PCA directly in the new feature space is highly inefficient; consequently, the kernel method is introduced to simplify the calculation by replacing $\boldsymbol{\Phi(X)\Phi(X)^T}$ with $\boldsymbol{K}$ [27]. Thus, there is no need to calculate $\Phi(X)$. To illustrate this process, $\boldsymbol{\Phi(X)}$ should be centered (4), i.e., $\sum_{i=1}^{t} \boldsymbol{\Phi(x_i)} = 0$, to compute the covariance matrix.

$$\bar{\boldsymbol{\Phi}}(\boldsymbol{X}) = \boldsymbol{\Phi(X)} - \frac{1}{N}\sum_{i}^{t} \boldsymbol{\Phi(x_i)} \tag{4}$$

The covariance matrix, which captures the correlation between the variables [28], can be computed as:

$$\bar{\boldsymbol{\Sigma}}_{\phi(X)} = \frac{1}{N}\sum_{i=1}^{N} \bar{\boldsymbol{\Phi}}(\boldsymbol{x_i})\bar{\boldsymbol{\Phi}}(\boldsymbol{x_j})^T \tag{5}$$

The eigenvectors $\mathbf{A}$ of the covariance matrix $\Sigma_{\phi(X)}$ is described in equation (6), which represent the direction of the data variance.

$$\lambda\mathbf{A} = \bar{\boldsymbol{\Sigma}}_{\phi(X)}\mathbf{A} \tag{6}$$

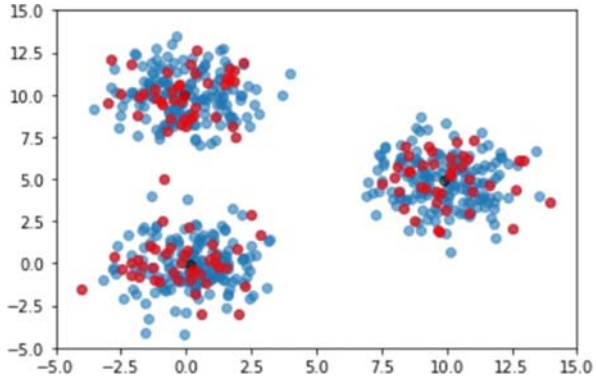


Fig. 4. An example of selecting representative data: blue points are the entire data, and the red points are the selected representative data.

From (5), $\mathbf{A} \in span\{\bar{\Phi}(x_1), \bar{\Phi}(x_2), \ldots, \bar{\Phi}(x_N)\}$ [29]. Hence, A can be stated as a linear combination of $\bar{\Phi}(x_i)$'s with coefficients α_i's:

$$\mathbf{A} = \sum_{i}^{N} \alpha_i \bar{\boldsymbol{\Phi}}(\boldsymbol{x_i}) \tag{7}$$

Substituting (7) and (5) into (6) yields (8) [29],

$$\lambda \sum_{i}^{N} \alpha_i \bar{\boldsymbol{\Phi}}(\boldsymbol{X_i}) = \frac{1}{N}\sum_{i=1}^{N} \bar{\boldsymbol{\Phi}}(\boldsymbol{x_i})\bar{\boldsymbol{\Phi}}(\boldsymbol{x_i})^T \sum_{i}^{N} \alpha_i \bar{\boldsymbol{\Phi}}(\boldsymbol{X_i})$$

$$\lambda\bar{\boldsymbol{\Phi}}(\boldsymbol{X})\boldsymbol{\alpha} = \frac{1}{N}\bar{\boldsymbol{\Phi}}(\boldsymbol{X})\bar{\boldsymbol{\Phi}}(\boldsymbol{X})^T\bar{\boldsymbol{\Phi}}(\boldsymbol{X})\boldsymbol{\alpha}$$

$$N\lambda\bar{\boldsymbol{\Phi}}(\boldsymbol{X})\bar{\boldsymbol{\Phi}}(\boldsymbol{X})^T\boldsymbol{\alpha} = \bar{\boldsymbol{\Phi}}(\boldsymbol{X})\bar{\boldsymbol{\Phi}}(\boldsymbol{X})^T\bar{\boldsymbol{\Phi}}(\boldsymbol{X})\bar{\boldsymbol{\Phi}}(\boldsymbol{X})^T\boldsymbol{\alpha} \tag{8}$$

Applying (3) into (8) yields the following equation [28],

$$N\lambda\bar{\boldsymbol{K}}\boldsymbol{\alpha} = \bar{\boldsymbol{K}}^{\boldsymbol{2}}\boldsymbol{\alpha} \longrightarrow N\lambda\boldsymbol{\alpha} = \bar{\boldsymbol{K}}\boldsymbol{\alpha}, \tag{9}$$

which is an n-dimensional eigenvalue problem. To satisfy the KPCA constraints, the norm of α is adjusted to $||\boldsymbol{\alpha}||^2 = \frac{1}{\lambda}$. Finally, the transformed data is calculated as $X_{trans} = \boldsymbol{\alpha}\bar{\boldsymbol{K}}$. A transformed data set of p dimensions can be achieved by selecting the first $p(p < N)$ columns of $\boldsymbol{\alpha}$ for computation.

2) Data Clustering: Following feature extraction from the dataset, the transformed data X_{trans} are classified into K groups according to their similarity. Spectral Clustering (SC) [30], which blends clustering and graph theory is leveraged for this purpose. Compared with the conventional clustering K-means, spectral clustering is more robust and has better grouping performance regardless of the data distribution.

SC combines graph theory and K-means, where graph theory pre-processes the data and K-means clusters the pre-processed data. Rather than directly clustering the data based on their value, SC groups data based on the eigenvectors of its Laplacian matrix L, which is obtained from the similarity matrix W.

Three methods are generally deployed to generate the similarity matrix W of the transformed dataset X_{trans}, : ϵ-neighborhood [31], k-nearest neighborhood [32], and the fully connected method [33]. In this paper, the fully-connected method is deployed, which assumes a fully-connected graph when constructing a similarity matrix due to relatively small

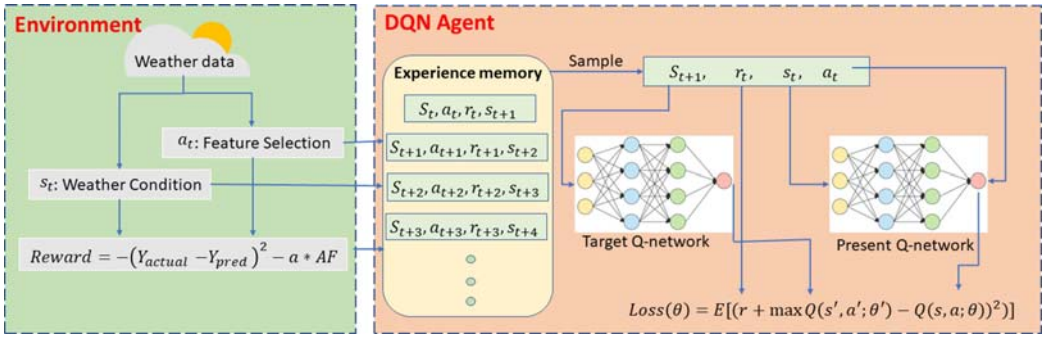


Fig. 5. Learning principle for the developed model.

data variance. Each W_{ij} in W represents the similarity between the samples i and j. Here, the Gaussian similarity function is used, as it assigns a higher weight to samples that are close [33]:

$$\boldsymbol{W_{ij}} = exp\left(-\frac{||\boldsymbol{X_i} - \boldsymbol{X_j}||^2}{2\sigma^2}\right), \tag{10}$$

where σ determines the distance of the data neighborhood. The Laplacian matrix L is calculated as [34],

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W} \tag{11}$$

where $D_{ii} = \sum_j W_{ij}$. The eigenvalues and their corresponding eigenvectors of the Laplacian matrix are represented as:

$$\boldsymbol{LA_i} = \lambda_i \boldsymbol{A_i} \tag{12}$$

Upon obtaining the eigenvectors and eigenvalues, the eigenvectors are sorted according to the value of the eigenvalues. The top K eigenvectors $\lambda_1, .., \lambda_k$, the number of which is similar to the predicted clusters and is determined later in the simulation section, are then chosen to build the feature matrix U, $U \in \mathbb{R}^{N \times K}$ for large-scale data. A high-dimensional feature matrix can be challenging to group effectively. Therefore, the selected K eigenvectors are the optimal solution to balancing the utilization of valuable information and reducing the processed eigenvectors' dimensionality [30]. The values in the feature matrix U are then grouped with K-means into K clusters, and the same grouping results are applied to X_{trans} [35]. To select the representative data of X_{trans}, the data in each cluster is equally divided into four groups (A-D) based on the Euclidean distance between the data and its centroid c_i. The first group (A) contains the top 25% of data closest to the centroid c_i, and the following groups are ranked similarly. The representative data for the ith cluster is selected as,

$$n_{s_i} \begin{cases} b_1, & Group\ A\ in\ the\ i_{t_h}\ cluster \\ b_2, & Group\ B\ in\ the\ i_{t_h}\ cluster \\ b_3, & Group\ C\ in\ the\ i_{t_h}\ cluster \\ b_4, & Group\ D\ in\ the\ i_{t_h}\ cluster \end{cases} \tag{13}$$

where, $b_1 > b_2 > b_3 > b_4$ and $b_1 + b_2 + b_3 + b_4 = n_i$. The data selection criterion in (13) implies that more samples are selected near the centroid. To ensure data diversity, some data farther from the centroid are also selected.

The transformed dataset X_{trans} contains the same information as the original weather dataset $X(x)$. Hence, the index of the representative data X_{trans}, which means where representative data is in X_{trans}, is applied to the original dataset $X(x)$ to obtain the final representative data. This process is also illustrated in Fig. 3. An example of selecting representative data from the original dataset is shown in Fig. 4, with blue dots representing the original dataset, and red points representing the chosen data.

C. Dynamic Feature Selection - Deep Reinforcement Learning

Although feature selection and clustering reduce the amount of data, all types of sensor measurements are still used to predict solar irradiance. There is a substantial difference across different weather conditions regarding the required sensor measurements. Not all variables are always necessary, and there is a potential to reduce the data volume for the predictions further. A novel dynamic feature selection methodology is developed here, extending DRL. DRL is a collection of goal-oriented algorithms that combine Deep Networks with RL. Although RL effectively solves a wide range of complex problems in different domains, only low-dimensional environments can be fully observed by RL [36]. This limitation hinders the application of RL to domains with complex environments. On the contrary, DRL can solve problems independent of the data dimensions. Since data dimensions could potentially be high for the problem at hand, Deep Q-Network (DQN), which incorporates Q-learning with Deep Neural Network (DNN), is leveraged here.

Q-learning is an off-policy DRL algorithm that discovers the optimum actions for a given state [37]. Upon defining the action and state space in an environment, an optimal action maximizes the expected value of the total reward in subsequent steps. The Q-value, denoted as $Q(State, Action)$, is the evaluation standard for each action. The action with the highest Q-value is generally regarded as the optimal action for the current state. The Q-value can be formulated as [38],

$$Q^{new}(s_t, a_t) = Q^{old}(s_t, a_t) + \alpha \times (r_t + \gamma \times maxQ(s_{t+1}, a) - Q^{old}(s_t, a_t)) \tag{14}$$

The computed Q-values are stored in a table known as the Q-table, and the maximum expected Q-value for each state could be directly obtained from the Q-table. In other words, Q-learning agents have prior knowledge about the optimal action under each state based on the Q-table. However, the number of actions and states in a high-dimensional problem can be infinite. In this study, the states, that is, the weather conditions, are countless. Consequently, Q-learning or RL is not a proper method for the learning task at hand and is significantly challenging to handle a state that was not trained when developing the RL

model. DQN can overcome the challenge mentioned above by incorporating Q learning with neural networks. The DNN component, which consists of two neural networks with the same structure, stores the associated Q-values by estimating a function $f(S, A) = Q(State, Action)$ [39]. Thus, DQN is not limited by the number of states and can estimate optimal actions even for untrained states. The defined actions and states are the inputs to DNN. Due to correlations between sequential inputs, directly learning from a succession of actions and states causes DNN to fall into local minima. Consequently, a mechanism known as experience replay is introduced to overcome this problem [40]. Upon acquiring a large number of Q-values with actions and states, and to make the input to DNN nonsequential, a fixed set of actions and states are randomly chosen as the input to the DNN model.

The output of the first DNN referred to as the target Q-network, is used as labels to calibrate the parameters of a second DNN, namely the present Q-network. The parameters of the present Q-network can be calibrated with a Loss Function represented as [41],

$$Loss_i(\theta) = E_{s,a}[(r + \gamma \max_{a'} Q(s,' a'; \theta_i^-) - Q(s, a; \theta_i))^2] \tag{15}$$

Upon differentiating the loss function, as shown in (16), the gradient descent method calibrates the parameters in the present Q-network [39].

$$\nabla Loss_i(\theta_i) = E_{s,a}[(r + \gamma \max_{a'} Q(s,' a'; \theta_i^-) - Q(s, a; \theta_i)) \times \nabla_{\theta_i} Q(s, a; \theta_i)] \tag{16}$$

$$\theta = \theta - \beta \nabla L(\theta) \tag{17}$$

For each P step, which is predetermined based on performance, the parameters in the target Q-network are replaced with the parameters in the present Q-network to update the learning results. This process is repeated until the reward converges, as summarized in **Algorithm 1**.

As shown in Fig. 5, the action is defined as the selected features and is represented by an N-digit binary number, where each bit represents a specific feature type. For example, the action is defined as the selected features and is represented by an N-digit binary number, where each bit represents a specific feature type. For example, the action '1010000000' indicates that only the first and third features have been selected for prediction among the ten available features. The state is defined as the numerical value of the ten categories of accessible meteorological data, such as specific temperature and humidity at the time of the forecast. Each state is then formed as a $\{1 \times 10\}$ array. The reward function aims to improve the accuracy of the forecast with less data. Hence, it is formulated as,

$$Reward = -(Y_{actual} - Y_{pred})^2 - a \times AF \tag{18}$$

A trained DQN model yields optimal features that dynamically change based on different weather conditions.

Algorithm 1: Deep Q-Network with experience reply.

```
Initialize present Q-Network with random parameters θ
Initialize target Q-Network with random parameters θ⁻ = θ
Initialize the experience memory to D with capacity C
for episode = 1,2,...,M do
    Set initial observation state s_1
    Initialize the sequence of state s = {s_1}
    for t = 1,2,...T do
        With probability ε select a random action a_t
        Otherwise, select a_t = argmax(s_t, a; θ)
        Execute selected action a_t
        Observe reward r_t and new state s_{t+1}
        Store transition (s_t, a_t, r_t, s_{t+1}) in D
        Sample a C capacity random minibatch of
        transitions (s_j, a_j, r_j, s_{j+1}) from D
        if episode ends at step j+1 then
            Set y_j = r_j
        else Set y_j = r_j + γ × maxQ(s_{j+1}, a'; θ⁻)
        end if
        Perform the gradient descent procedure on
        (y_j − Q(s_j, a_j; θ))², updating the parameters θ
        in the present Q-Network
        For every P step, replace θ⁻ with θ
    end for
end for
```

D. Extreme Gradient Tree Boosting Forecasting Model

The identified optimal features are deployed by the Extreme Gradient Tree Boosting (XGBoost) [42] algorithm to generate the final solar irradiance forecasts. XGBoost is a scalable machine learning algorithm for tree boosting, commonly used to solve regression and classification problems [42]. The advantage of XGBoost over other common methods (such as NN) is that it is robust and adaptive to diverse and dynamic data found in real world. The major limitation of XGBoost is that it is sensitive to data outliers, which affects the performance of XGBoost. However, the outlier data in this study are filtered out by selecting representative data from the entire dataset. In other words, XGBoost and the developed data selection model enhance each other.

Using the optimal selected features, the initial predicted value from XGBoost is calculated as $\hat{y} = \sum_{k=1}^{n} f_k(x)$, $f_k(x)$ is the k_{th} independent regression tree with structure q and leaf weights w. An iterative method is then used to optimize the result by minimizing the following objective [42]:

$$\begin{cases} \mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y_i}^{(t-1)} + f_t(x_i)) + \Omega(f_t) \\ \Omega(f_t) = \epsilon T_t + \frac{1}{2}\mu ||w_t||^2 \end{cases} \tag{19}$$

where the first term $l(y_i, \hat{y_i}^{(t-1)} + f_t(x_i))$ refers to the loss function over the training set, and the second term $\Omega(f_t)$ is the penalty for the model's complexity [43]. However, all the tree structures in $f(\cdot)$ are impossible to be listed for optimization. Using Taylor expansion and the concept of greedy algorithm to

split the tree, (19) is represented as (20) after splitting the tree nodes [42].

$$\mathcal{L}_{split} = \frac{1}{2}\left[\frac{(\sum_{i\in\mathcal{I}_L} g_i)^2}{\sum_{i\in\mathcal{I}_L} h_i + \mu} + \frac{(\sum_{i\in\mathcal{I}_R} g_i)^2}{\sum_{i\in\mathcal{I}_R} h_i + \mu} - \frac{(\sum_{i\in\mathcal{I}} g_i)^2}{\sum_{i\in\mathcal{I}} h_i + \mu}\right] - \epsilon \quad (20)$$

where $g_i = \partial_{\hat{y_i}^{(t-1)}} l(y_i, \hat{y_i}^{(t-1)})$ and $h_i = \partial^2_{\hat{y_i}^{(t-1)}} l(y_i, \hat{y_i}^{(t-1)})$. Equation (20) is used to find the best node to divide the tree. The objective $\mathcal{L}$ is then easily optimized to determine the leaf weights of the entire tree.

IV. Simulation and Summary

The developed DQN-based model is assessed in this section with real-world data. The platform for simulation has an Intel(R) Xeon(R) CPU @ 2.20 GHz with 13 GB RAM. Results are obtained and compared with traditional forecasting methods without applying the developed model. Prediction is performed based on all the original weather data when the developed model is not used. The robustness of the developed predictive model is also examined under extreme weather conditions, such as overcast cloudy days and heavy rainy days.

A. Data Characteristics and Evaluation Criteria

The weather data used for simulation were obtained from the Open Weather's database [44] and were collected from January 1, 2019 to December 31st 2021, at Seattle, Washington. The data set contains hourly weather data from different sensors. Sensors collect hourly temperature, dew points, feel-like temperature, air pressure, relative humidity, average wind speed, wind degree, cloud cover, and visibility. Data from January 1, 2019 to December 31, 2020 are treated as the training dataset, and data from January 1, 2021 to December 31, 2021 are considered the test dataset. Night data is filtered out in both the training and the test process, as only daytime predictions have practical meaning.

To evaluate a forecast, several standard statistical metrics, i.e., the Root Mean Square Error (RMSE): $\sqrt{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2/n}$ [45], the normalized Root Mean Square Error (nRMSE): $1/(Y_{max} - Y_{min}) \cdot \sqrt{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2/n}$ [45], the Mean Absolute Percent Error (MAPE): $100\%/n \cdot \sum_{i=1}^{n} |(Y_i - \hat{Y}_i)/Y_i|$ [45], and R^2 score: $1 - (\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2)/(\sum_{i=1}^{n}(Y_i - \bar{Y})^2)$ [46] are used.

B. Benchmark

1) Persistence Ensemble: A traditional time series forecasting method, that is, the persistence ensemble method [47], is introduced as the first benchmark. The results of day-ahead Global Horizontal Irradiance (GHI) prediction from the Persistence method are directly obtained from National Oceanic and Atmospheric Administration (NOAA) SOLRAD Seattle station [48] for comparison.

2) Neural Network: A common approach for forecasting, i.e., Feed Forward Neural Network (FFNN) [49], is introduced as the second benchmark. The developed FFNN generates forecasts using the optimal features determined by the well-trained DQN model, is built with training data, and the near-optimal inner parameters are obtained by cross-validation.

3) Autoencoder Long-Short-Term Memory Network (AE-LSTM): A recent approach for solar forecasting, which is illustrated in [19], is introduced as the second benchmark. The top six relevant features are first identified using the Root Mean Squared Euclidean Distance Difference (RMSEDD), as recommended in [19]. The relevant features are utilized to train the AE-LSTM model, which is then optimized by an optimizer. The loss function is defined as the MSE metric, the activation function is sigmoid function, and the near-optimal inner parameters are obtained by cross-validation.

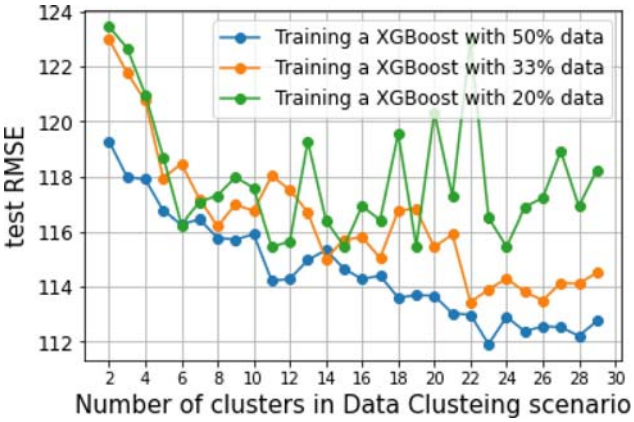


Fig. 6. Variations of testing RMSE when increasing the number of clusters in the Data Clustering step

TABLE I
Comparison of Test Error When Training XGBoost With Different Percentage of Representative Data

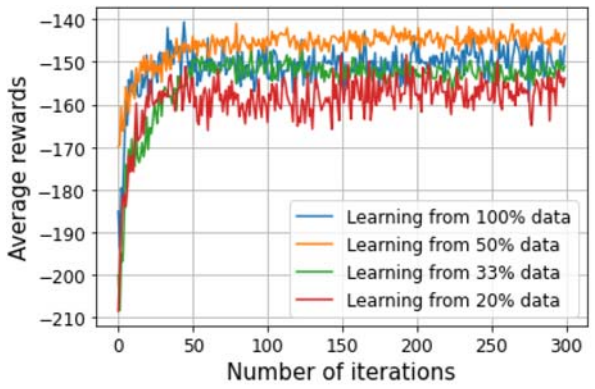
Train a XGBoost with	MAPE (%)	RMSE	nRMSE	R^2
100% data*	7.6937	113.473	0.1138	0.8094
50% data	7.646	111.907	0.1122	0.8135
33% data	7.8299	115.020	0.1153	0.805
20% data	7.9090	115.508	0.1158	0.7980

*Training a XGBoost with 100% data refers to the case without applying data selection step.

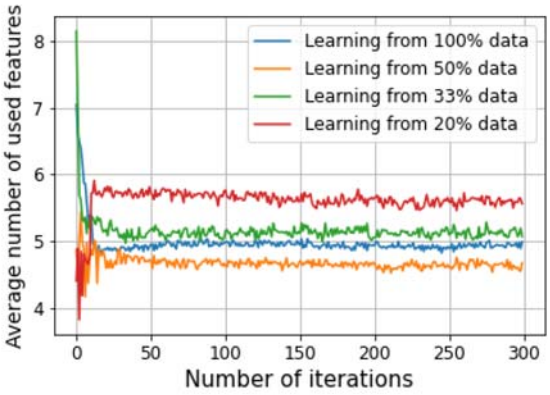
C. The Selection of Representative Data

First, the optimal hyperparameters of XGBoost are found using training data and cross-validation. The selection of representative data is evaluated in Fig. 6. This figure depicts the impact of the number of clusters on the performance of the representative data. Three different cases are investigated here, i.e., 50%, 33%, and 20% of training data are selected as representative data. The y-axis shows the RMSE of the forecasts based on the test data.

From Fig. 6, it is seen that the optimal number of clusters for the case with 50% of data is 23. For 33% and 20% of data cases, that number is 22 and 23, respectively. The best performances using different percentages of representative data are summarized in Table I. These results demonstrate that training



(a) Variations of the reward



(b) Variations of the average number of used features

Fig. 7. The reward variations, and the average number of used features in every iteration when learning from different percentages of training data.

a forecast model only with appropriate representative data, i.e., 50% data case, increases the accuracy.

D. The Learning Performance of the Dynamic Feature Selection Step

The four different representative datasets in Table I are used by the developed DQN model to learn and select the optimal features, which the XGBoost then utilizes to generate forecasts. Variations in the reward value and the average number of features used for forecasting in every iteration of the four aforementioned cases are provided in Fig. 7.

As seen from the figure, when 20% and 33% of the training data are selected, the learning curves exhibit more oscillations, and the rewards eventually converge to a relatively lower value. When 50% of the data are chosen, the learning curve becomes smoother, and the reward converges at a higher value with less used features for forecasting. Compared with baseline, where the training data is used entirely, the reward for the 50% case exhibits better performance. The baseline case exhibits overfitting, a problem that is not observed when using half of the data. With 50% of the data, the average number of optimal features converges to around five, much less than the total number of variables, i.e., ten. These results also corroborate the initial hypothesis: Learning from representative data can improve the effectiveness and efficiency of the DQN model, resulting in better learning results. The forecasting performance of the four cases is summarized in Table II.

TABLE II
COMPARISON OF TEST ERROR WHEN TRAINING THE DEVELOPED HYBRID MODEL WITH DIFFERENT PERCENTAGE OF REPRESENTATIVE DATA

DQN learns from	MAPE (%)	RMSE	nRMSE	R^2
100% data*	7.6142	112.058	0.1124	0.8142
50% data	7.5646	111.044	0.1114	0.8175
33% data	7.730	112.996	0.1133	0.811
20% data	7.8205	114.364	0.11474	0.8064

TABLE III
COMPARISON OF THE TEST ERROR WHEN FORECASTING WITH DIFFERENT CASES

	MAPE (%)	RMSE	nRMSE	R^2
XGBoost	7.6937	113.473	0.1138	0.8094
KPCA+XGBoost	10.27	140.639	0.1411	0.7073
SC+XGBoost	7.98	116.23	0.1166	0.8
KPCA+SC+XGBoost	7.646	111.907	0.1122	0.8135
DQN+XGBoost	7.6142	112.058	0.1124	0.8142
Developed model*	7.5646	111.044	0.1114	0.8175

*Developed model refers to KPCA+SC+DQN+XGBoost

As shown in Table II, the forecast error for the case where DQN learns from 50% of the data is generally lower in the other three cases. In this case, the number of features used for the forecast is less than half of the total number of features, according to Fig. 7(b). Hence, a more accurate forecast is achieved by only using around 25% (50% $\times$ 50% $=$ 25%) of the training data. The well-trained DQN is thus used to determine dynamic optimum features for future solar irradiance forecasting. Forecasting with optimum features reduces computing time, resulting in faster prediction results. The scenario utilizing optimum features takes 0.2 seconds to generate the prediction for day-ahead hourly predictions, whereas the scenario using all available data takes 0.28 seconds. The developed model's calculation time for generating predictions is almost the same as the case of benchmark AE-LSTM, which takes 0.22 seconds. Another benchmark model Persistence Ensemble uses 0.004 seconds to yield online forecasts. Despite the fact that Persistence Ensemble has faster computing time than our developed model due to its simple structure, forecasts generated by the developed model are substantially more accurate than those generated by Persistence Ensemble.

Furthermore, the impact of different components on the developed model is reflected in Table III. As shown in Table III, the DQN component contributes the most to an accurate forecast, since using DQN only to select optimal features (DQN+XGBoost) can increase the accuracy of the forecast. On the other hand, only clustering or transforming the data to reduce the data volume (SC+XGBoost or KPCA+XGBoost) will not lead to improved forecast accuracy. Combining KPCA with SC to pre-process the raw dataset can slightly enhance the forecasting performance, supporting the stated hypothesis: applying feature extraction to the raw dataset leads to better clustering and more accurate forecasts. The developed model, which refers to the "KPCA+SC+DQN+XGBoost" case, has

TABLE IV
COMPARISON OF THE FORECASTING PERFORMANCE OF THE DEVELOPED MODEL WITH BENCHMARK

Model	MAPE (%)	RMSE	nRMSE	R^2
Persistence Ensemble	14.414	206.283	0.2069	0.4022
AE-LSTM	7.73	113.54	0.114	0.809
Pearson+XGBoost	8.296	121.54	0.122	0.7814
KPCA+SC+DQN+FFNN	8.763	125.072	0.1255	0.7685
Developed model*	7.5646	111.044	0.1114	0.8175

*Developed model refers to KPCA+SC+DQN+XGBoost

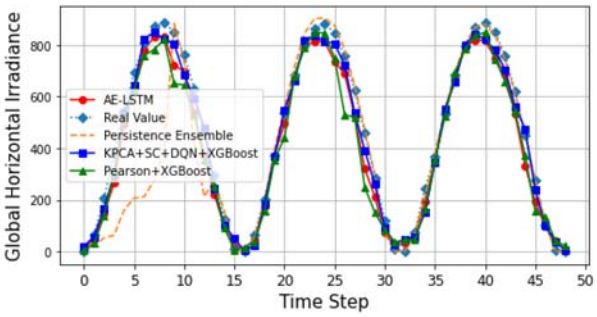
the lowest prediction error among all instances examined in Table III, indicating the combination of KPCA, SC, and DQN increases prediction accuracy even further.

E. Comparison With the Benchmark

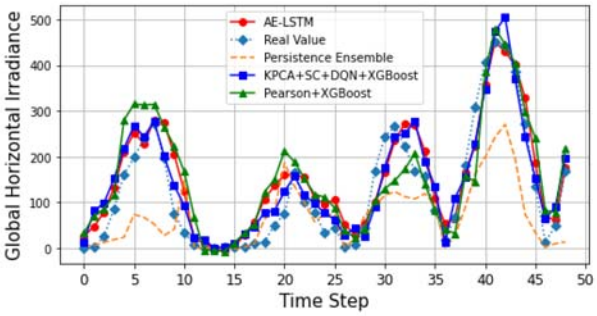
The test data are collected in Seattle, WA, from January 1 to December 31, 2021. The weather conditions during this period are relatively diverse, with more than 60% days being cloudy, rainy, and snowy. Therefore, the solar irradiance forecast model developed is thoroughly evaluated in extreme weather conditions. The forecasting performance of the hybrid model developed is compared to the benchmark methods, as shown in Table IV. The forecasts from the Persistence Ensemble method are directly obtained from National Oceanic and Atmospheric Administration (NOAA) SOLRAD Seattle station [48]. Other models are trained with training data and optimized by cross-validation. Additionally, a traditional feature selection method, i.e., Pearson correlation coefficient, is also chosen for comparison. Pearson correlation coefficient (21) is a statistical measure that indicates how related two variables are [50].

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \cdot \sigma_Y} \tag{21}$$

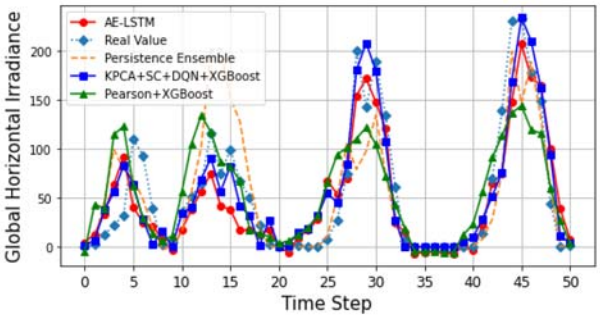
The top five correlated features are selected and applied to train an XGBoost model with cross-validation. The errors shown in Table IV refer to the test data. The forecasting performance of the developed model outperforms the benchmark methods, such as Persistence Ensemble and AE-LSTM methods. Persistence ensemble generates forecasts based on past days' real solar irradiance, and forecast accuracy can only be guaranteed for consecutive days with similar weather conditions. Nevertheless, such circumstances are often not realized in the real world, which is the primary reason for Persistence Ensemble's poor forecasting performance. Furthermore, the AE-LSTM model or traditional methods based on feature selection utilize constant features for forecasting, which restricts the robustness of the predictive model as different features are appropriate for different weather conditions. On the contrary, the developed model dynamically selects optimal features used for forecasting at each point in time, determined by the well-trained DQN model under various weather conditions. The developed model selects the most appropriate features for generating forecasts, thus outperforming the analyzed traditional methods by dynamically selecting fewer features while ensuring a more accurate forecast. In short, the hybrid solar irradiance forecasting model produced more accurate forecasts with fewer data.



(a) Three consecutive sunny days in summer



(b) One cloudy day followed by two consecutive rainy days and a sunny day in autumn.

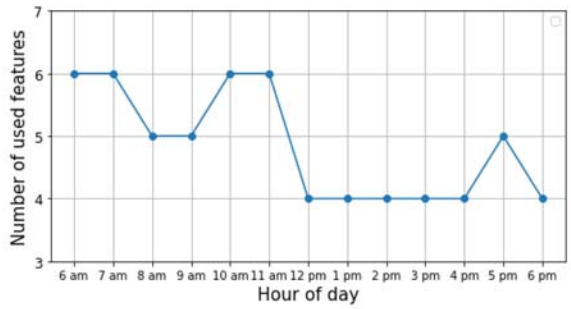


(c) Two consecutive rainy days followed by two consecutive cloudy days in winter

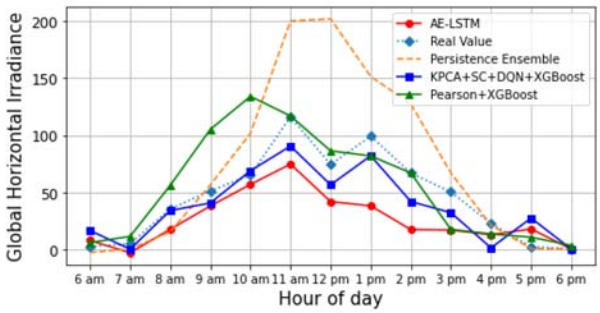
Fig. 8. Comparison of forecasts from different models under diverse weather conditions.

F. Case Studies

The forecasted results for various weather conditions are shown in Fig. 8. GHI values shown in Fig. 8 are daytime values, as nighttime data are filtered out. Fig. 8(a) demonstrates the forecasts for three consecutive sunny days in summer. As seen from the figure the Persistence Ensemble method generates the most accurate forecasts for the last two days, as this method has been proven to be perfectly suitable for consecutive sunny days [19]. However, the first day's forecast from the Persistence Ensemble method is quite inaccurate since the day before the first day is a cloudy day. In contrast, the developed model is much more stable while maintaining high forecast accuracy. Fig. 8(b) shows the forecasts for one cloudy day followed by two consecutive rainy days and a sunny day in the Fall. The forecasts of the last sunny day from Persistence Ensemble are



(a) Variations of used features for forecasting by using the developed model for a rainy day



(b) Solar irradiance forecasts for a rainy day

Fig. 9. Comparison of forecasts from different models for a rainy day.

TABLE V
OPTIMAL FEATURES USED FOR PREDICTIONS FOR THE RAINY DAY

Time	Types of used features for predictions.
6-7 am	Air temperature, Feel-like temperature, Pressure, Humidity,Wind degree, and Cloud cover
8-9 am	Feel-like temperature, Humidity, Wind speed, Cloud cover, and Visibility
10-11 am	Dew point, pressure, Humidity, Wind speed, Cloud cover, and Visibility
12-4 pm	Feel-like temperature, Humidity, Wind speed, and Cloud cover

significantly underestimated since they are based on real GHI from the previous several days, which are generally low due to the unfavorable weather conditions. Despite the fact that other approaches produce reasonable forecasts, the forecasts from the developed model are the most accurate after careful examination. Fig. 8(c) shows the forecasts for two consecutive rainy days followed by two consecutive cloudy days. Even on the analyzed cloudy days in the winter, when reliable prediction is challenging, solar irradiance can still be accurately forecasted using the developed model. These case studies demonstrated that the developed DQN model significantly adapts to rapid changes in all weather conditions, and remarkably outperforms the traditional forecasting methods, such as the persistence ensemble method.

1) An Example of the Change in Features for a Daily Forecast: Fig. 9 shows the variation of the features selected for forecasting using the model developed during a rainy day. As illustrated in Table V, the types of features used to forecast change throughout the day. These observations demonstrate that the number and types of optimal features selected by the developed model dynamically change from hour to hour.

V. CONCLUSION

This paper presents a DQN-based forecasting model to determine the optimal variables for accurate solar irradiance forecasting. The objective of the developed model is to build a predictor that adapts to varying weather conditions and is particularly useful when there is limited access to data for solar generation prediction. Case studies using real-world data have demonstrated that the developed model significantly decreases the volume of data required for accurate solar irradiance forecasting under various weather conditions while slightly increasing the forecast accuracy. The reduction of data volume used for prediction increases computing efficiency and reduces storage costs. The developed method is model-free and can be applied in other applications, such as wind forecasting, load forecasting, or forecast applications in other disciplines, where reducing the volume of data to be processed is of interest. This method can be beneficial to on-board processing applications, where the computational power and communication constraints limit the volume of data to be processed.

Solar energy under extreme weather conditions is unstable and prone to constant fluctuations, making accurate prediction extremely difficult. This research can be extended to further quantify the correlation between forecasted weather data and actual data and incorporate the uncertainty of forecast models with DRL to produce a generation uncertainty quantification framework under various weather conditions.

REFERENCES

[1] M. Castangia, A. Aliberti, L. Bottaccioli, E. Macii, and E. Patti, "A compound of feature selection techniques to improve solar radiation forecasting," *Expert Syst. Appl.*, vol. 178, 2021, Art. no. 114979.
[2] H. Liu, C. Yu, C. Yu, C. Chen, and H. Wu, "A novel axle temperature forecasting method based on decomposition, reinforcement learning optimization and neural network," *Adv. Eng. Inform.*, vol. 44, 2020, Art. no. 101089.
[3] F. Douak, F. Melgani, and N. Benoudjit, "Kernel ridge regression with active learning for wind speed prediction," *Appl. Energy*, vol. 103, pp. 328–340, 2013.
[4] H. Liu, C. Yu, H. Wu, Z. Duan, and G. Yan, "A new hybrid ensemble deep reinforcement learning model for wind speed short term forecasting," *Energy*, vol. 202, 2020, Art. no. 117794.
[5] C. Feng, M. Sun, and J. Zhang, "Reinforced deterministic and probabilistic load forecasting via *q*-learning dynamic model selection," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1377–1386, Mar. 2020.
[6] C.-F. Chien, Y.-S. Lin, and S.-K. Lin, "Deep reinforcement learning for selecting demand forecast models to empower industry 3.5 and an empirical study for a semiconductor component distributor," *Int. J. Prod. Res.*, vol. 58, no. 9, pp. 2784–2804, 2020.
[7] W. Fan, K. Liu, H. Liu, P. Wang, Y. Ge, and Y. Fu, "Autofs: Automated feature selection via diversity-aware interactive reinforcement learning," in *Proc. IEEE Int. Conf. Data Mining*, 2021, pp. 1008–1013.
[8] M. Kroon and S. Whiteson, "Automatic feature selection for model-based reinforcement learning in factored MDPs," in *Proc. IEEE Int. Conf. Mach. Learn. Appl.*, 2009, pp. 324–330.
[9] S. M. H. Fard, A. Hamzeh, and S. Hashemi, "Using reinforcement learning to find an optimal set of features," *Comput. Math. with Appl.*, vol. 66, no. 10, pp. 1892–1904, 2013.
[10] D. Yang, J. Kleissl, C. A. Gueymard, H. T. Pedro, and C. F. Coimbra, "History and trends in solar irradiance and pv power forecasting: A preliminary assessment and review using text mining," *Sol. Energy*, vol. 168, pp. 60–101, 2018.

[11] R. Perez, S. Kivalov, J. Schlemmer, K. Hemker Jr, D. Renné, and T. E. Hoff, "Validation of short and medium term operational solar radiation forecasts in the us," *Sol. Energy*, vol. 84, no. 12, pp. 2161–2172, 2010.
[12] A. Hammer, D. Heinemann, E. Lorenz, and B. Lückehe, "Short-term forecasting of solar radiation: A statistical approach using satellite data," *Sol. Energy*, vol. 67, no. 1–3, pp. 139–150, 1999.
[13] M. Ueshima, T. Babasaki, K. Yuasa, and I. Omura, "Examination of correction method of long-term solar radiation forecasts of numerical weather prediction," in *Proc. IEEE 8th Int. Conf. Renewable Energy Res. Appl.*, 2019, pp. 113–117.
[14] C. Voyant et al., "Machine learning methods for solar radiation forecasting: A review," *Renewable Energy*, vol. 105, pp. 569–582, 2017.
[15] F. Davò, S. Alessandrini, S. Sperati, L. Delle Monache, D. Airoldi, and M. T. Vespucci, "Post-processing techniques and principal component analysis for regional wind power and solar irradiance forecasting," *Sol. Energy*, vol. 134, pp. 327–338, 2016.
[16] H. Long, Z. Zhang, and Y. Su, "Analysis of daily solar power prediction with data-driven approaches," *Appl. Energy*, vol. 126, pp. 29–37, 2014.
[17] C. Paoli, C. Voyant, M. Muselli, and M.-L. Nivet, "Forecasting of preprocessed daily solar radiation time series using neural networks," *Sol. Energy*, vol. 84, no. 12, pp. 2146–2160, 2010.
[18] Z. Zhen et al., "Deep learning based surface irradiance mapping model for solar PV power forecasting using sky image," *IEEE Trans. Ind. Appl.*, vol. 56, no. 4, pp. 3385–3396, Jul./Aug. 2020.
[19] Y. Zhang, C. Qin, A.K. Srivastava, C. Jin, and R. K. 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.
[20] J. Yan et al., "Frequency-domain decomposition and deep learning based solar pv power ultra-short-term forecasting model," *IEEE Trans. Ind. Appl.*, vol. 57, no. 4, pp. 3282–3295, Jul./Aug. 2021.
[21] Z. Si, Y. Yu, M. Yang, and P. Li, "Hybrid solar forecasting method using satellite visible images and modified convolutional neural networks," *IEEE Trans. Ind. Appl.*, vol. 57, no. 1, pp. 5–16, Jan./Feb. 2020.
[22] C. Lyu, S. Basumallik, S. Eftekharnejad, and C. Xu, "A data-driven solar irradiance forecasting model with minimum data," in *Proc. IEEE Texas Power Energy Conf.*, 2021, pp. 1–6.
[23] S. Ryu, H. Choi, H. Lee, and H. Kim, "Convolutional autoencoder based feature extraction and clustering for customer load analysis," *IEEE Trans. Power Syst.*, vol. 35, no. 2, pp. 1048–1060, Mar. 2020.
[24] T. T. Nguyen, P. Krishnakumari, S. C. Calvert, H. L. Vu, and H. Van Lint, "Feature extraction and clustering analysis of highway congestion," *Transp. Res. Part C: Emerg. Technol.*, vol. 100, pp. 238–258, 2019.
[25] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, *Feature Extraction: Foundations and Applications*, vol. 207. Berlin, Germany: Springer, 2008.
[26] I. Jaffel, O. Taouali, M. F. Harkat, and H. Messaoud, "Kernel principal component analysis with reduced complexity for nonlinear dynamic process monitoring," *Int. J. Adv. Manuf. Technol.*, vol. 88, no. 9-12, pp. 3265–3279, 2017.
[27] W. Zheng, C. Zou, and L. Zhou, "An improved algorithm for kernel principal component analysis," *Neural Process. Lett.*, vol. 22, no. 1, pp. 49–56, 2005.
[28] K. I. Kim, K. Jung, and H. J. Kim, "Face recognition using kernel principal component analysis," *IEEE Signal Process. Lett.*, vol. 9, no. 2, pp. 40–42, Feb. 2002.
[29] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *Proc. Int. Conf. Artif. Neural Netw.*, Springer, 1997, pp. 583–588.
[30] U. V. Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.
[31] S. Pourbahrami, L. M. Khanli, and S. Azimpour, "A novel and efficient data point neighborhood construction algorithm based on apollonius circle," *Expert Syst. With Appl.*, vol. 115, pp. 57–67, 2019.
[32] K. K. Sharma and A. Seal, "Spectral embedded generalized mean based k-nearest neighbors clustering with s-distance," *Expert Syst. with Appl.*, vol. 169, 2021, Art. no. 114326.
[33] J. Liu and J. Han, "Spectral clustering," in *Data Clustering*, Chapman and Hall/CRC, 2018, pp. 177–200.
[34] R. Chandra et al., "Forecasting trajectory and behavior of road-agents using spectral clustering in graph-LSTMs," *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 4882–4890, Jun. 2020.
[35] R. Azimi, M. Ghayekhloo, and M. Ghofrani, "A hybrid method based on a new clustering technique and multilayer perceptron neural networks for hourly solar radiation forecasting," *Energy Convers. Manage.*, vol. 118, pp. 331–344, 2016.
[36] Y. Ding et al., "Intelligent fault diagnosis for rotating machinery using deep q-network based health state classification: A deep reinforcement learning approach," *Adv. Eng. Inform.*, vol. 42, 2019, Art. no. 100977.
[37] J. Fan, Z. Wang, Y. Xie, and Z. Yang, "A theoretical analysis of deep Q-learning," in *Proc. Learn. Dyn. Control*, 2020, pp. 486–489.
[38] C. J. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, no. 3–4, pp. 279–292, 1992.
[39] V. Mnih et al., "Human-level control through deep reinforcement learning," *Natural*, vol. 518, no. 3–7540, pp. 529–533, 2015.
[40] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in *Proc. Int. Conf. Learn. Representations*, Puerto Rico, 2016.
[41] Y. Du, F. Zhang, and L. Xue, "A kind of joint routing and resource allocation scheme based on prioritized memories-deep q network for cognitive radio ad hoc networks," *Sensors*, vol. 18, no. 7, 2018, Art. no. 2119.
[42] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 785–794.
[43] R. Mitchell and E. Frank, "Accelerating the XGboost algorithm using gpu computing," *PeerJ Comput. Sci.*, vol. 3, 2017, Art. no. e127.
[44] D. Ukolov and O. Ukolov, "Openweathermap api." OpenWeather Ltd. [Online]. Available: https://openweathermap.org/
[45] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)?–arguments against avoiding RMSE in the literature," *Geoscientific Model Develop.*, vol. 7, no. 3, pp. 1247–1250, 2014.
[46] M. Kramer, "R2 statistics for mixed models," in *Proc. Conf. Appl. Statist. Agriculture*, 2005, vol. 17, pp. 148–160.
[47] H. Feddersen and U. Andersen, "A method for statistical downscaling of seasonal ensemble predictions," *Tellus A: Dyn. Meteorol. Oceanogr.*, vol. 57, no. 3, pp. 398–408, 2005.
[48] T. S. station in seattle, "National oceanic and atmospheric administration solrad seattle washington," [Online]. Available: https://gml.noaa.gov/grad/solrad/sea.html
[49] M. H. Sazli, "A brief review of feed-forward neural networks," *Commun. Fac. Sci. Univ. Ankara Ser. A2-A3 Phys. Sci. Eng.*, vol. 50, no. 01, 2006.
[50] H. Mo, H. Sun, J. Liu, and S. Wei, "Developing window behavior models for residential buildings using xgboost algorithm," *Energy Buildings*, vol. 205, 2019, Art. no. 109564.