e-Prime - Advances in Electrical Engineering, Electronics and Energy 10 (2024) 100853

Contents lists available at ScienceDirect

e-Prime - Advances in Electrical Engineering, Electronics and Energy

journal homepage: www.elsevier.com/locate/prime

Modern deep neural networks for Direct Normal Irradiance forecasting: A classification approach

Muhammad Saud Ul Hassan [a], Kashif Liaqat [a,*], Laura Schaefer [a], Alexander J. Zolan [b]

[a] Department of Mechanical Engineering, Rice University, Houston TX, 77005, United States
[b] Thermal Energy Systems, National Renewable Energy Laboratory, Golden CO, 80401, United States

ARTICLE INFO

Keywords:
Direct Normal Irradiance
Concentrated solar power
Deep neural networks
Recurrent neural networks
Transformers

ABSTRACT

The escalating energy demand and the adverse environmental impacts of fossil-fuel use necessitate a shift towards cleaner and renewable alternatives. Concentrated Solar Power (CSP) technology emerges as a promising solution, offering a carbon-free alternative for power generation. The efficiency and profitability of CSP depend on the Direct Normal Irradiance (DNI) component of solar radiation; hence, accurate DNI forecasting can help optimize CSP plants' operations and performance. The unpredictable nature of weather phenomena, particularly cloud cover, introduces uncertainty into DNI projections. Existing DNI forecasting models use meteorological factors, which are both challenging to estimate numerically over short prediction windows and expensive to model through data at a sufficiently high spatial and temporal resolution. This research addresses the challenge by presenting a novel approach that formulates DNI prediction as a multi-class classification problem, departing from conventional regression-based methods. The primary objective of this classification framework is to identify optimal periods aligning with specific operational thresholds for CSP plants, contributing to enhanced dispatch optimization strategies. We model the DNI classification problem using four advanced deep neural networks – rectified linear unit (ReLU) networks, 1D residual networks (ResNets), bidirectional long short-term memory (BiLSTM) networks, and transformers – achieving accuracies up to 93.5% without requiring meteorological parameters.

1. Introduction

Global demand for energy has been increasing rapidly, and as a result, the rate of fossil fuel depletion is increasing [1]. Moreover, dependence on fossil fuels has caused significant harm to the environment [2], with the global temperature now increasing at an alarming rate of 0.2 °C per decade [3]. These factors, coupled with reduced costs have led the energy market to embrace renewable and cleaner energy alternatives. Among the various choices available, concentrated solar power (CSP) technology is seeing rapid adoption [4] due to its dependable, secure, efficient, and environmentally-friendly approach to power generation [5]. Compared to other conventional renewable technologies, such as photovoltaic and wind energy, CSP allows for hybridization as well as storage of solar energy as heat [6]. The integration of thermal storage makes CSP dispatchable, and rectifies production fluctuations resulting from the inherently variable nature of solar radiation [7], thus helping to stabilize and control power output from the plant [5].

Solar radiation, known as *global horizontal irradiance*, is comprised of two main components: *direct normal irradiance* (DNI) and diffused irradiance. DNI represents the solar irradiance received directly from the sun on a surface normal to its path. CSP plants utilize DNI to generate electricity [8], as diffused irradiance cannot be efficiently concentrated [9]. Consequently, the operations, thermal performance, and economics of a CSP power plant depend on DNI and are affected by the resource variation. However, the unpredictable nature of weather phenomena, particularly cloud cover, makes it challenging to forecast DNI accurately. This introduces uncertainty into the projected output of CSP projects, creating issues for the integration, safety, and profitability of CSP initiatives [9].

Conventional methods and tools used for the design, simulation, and optimization of CSP power plants utilize historical DNI data in typical meteorological year (TMY) format to estimate the power output and system performance [10–13]. Additionally, DNI affects the excess flux distribution on the receiver, and hence its design and material safety of the receiver [14]. As a result, accurate forecasting of DNI is crucial to reliably estimate the productivity of CSP plants and for enhancing performance by coupling with other operational strategies

* Corresponding author.
E-mail addresses: ms18ig@my.fsu.edu (M. Saud Ul Hassan), Kashif.Liaqat@rice.edu (K. Liaqat), Laura.Schaefer@rice.edu (L. Schaefer), Alexander.Zolan@nrel.gov (A.J. Zolan).

https://doi.org/10.1016/j.prime.2024.100853
Received 12 May 2024; Received in revised form 7 November 2024; Accepted 18 November 2024
Available online 27 November 2024

such as dispatch optimization [15]. Integration of dispatch optimization with DNI forecasting has been shown to result in a significant increase in the annual revenue of a CSP power plant [16].

Previously, the DNI forecasting problem has only been formulated as a regression task. In this work, for the first time, we formulate DNI forecasting as a classification problem with target application in CSP plants. We discretize DNI values into classes corresponding to different potential operational modes of a CSP plant. For instance, a binary classifier can allow one to forecast whether there is enough DNI for a CSP plant to be operated safely or not. For more granularity, a ternary classifier can be deployed to forecast whether a plant can be efficiently and profitably operated on partial load, full capacity, or if it cannot be operated at all. Our multi-class classification framework lends to DNI forecasting models supporting operational schedules of arbitrary granularity. Furthermore, these classification models provide a probability distribution over the different operational modes of a CSP plant, which is a feature that regression-based models lack. Additionally, our classification-based treatment allows prior knowledge of DNI cutoffs for the different potential operational modes of a CSP plant to be injected into the forecasting model. This helps against the noise in the DNI data, which regression-based models are susceptible to. It is worth noting that this approach of formulating a regression problem as classification has been effective in other domains as well. For example, it is particularly popular in neural network models for predicting multi-modal distributions, such as vehicle orientation [17]. Moreover, studies over various datasets have demonstrated that classification algorithms can be superior to regression algorithms both in terms of accuracy and CPU time [18], inspiring machine learning algorithms for regression that are based on classification, such as the rule-based regression method presented by Weiss and Indurkhya [19]. While the primary novelty of our work is introducing the classification approach for DNI forecasting, our work also distinguishes itself from prior literature by not requiring meteorological parameters as input, which are extensively studied in existing research [20–22]. Since meteorological data is often difficult to acquire or expensive to model, we propose a direct mapping from time to DNI classes. This is effectively enabled by our multi-class classification approach for DNI forecasting, as regression-based models tend to be more sensitive to the loss of predictors.

The paper's outline is as follows: Section 2 reviews the previously proposed DNI forecasting methods, all of which require accurate estimates of cloud cover and other meteorological factors. Section 3 introduces our neural network models for DNI forecasting which do away with the need for meteorological data. These networks leverage our multi-class classification formulation of the DNI estimation problem, built on the fact that, in many applications of DNI forecasting to CSP plants, it suffices to assert whether DNI falls within specific bounds, rather than requiring an exact numerical value [15,16]. A thorough examination of our neural networks' performance on DNI prediction is provided in Section 4, where we demonstrate that they significantly outperform classical methods such as kernel machines. Finally, concluding remarks and future directions are presented in Section 5.

2. Literature review

The literature in this field discusses various numerical and data-based models for forecasting DNI over different time frames. Prior research has consistently emphasized the significant influence of cloudiness on the amount of solar radiation reaching the Earth's surface [23], and as a result, both numerical and data-based models aimed at DNI forecasting take into account a range of meteorological factors related to cloudiness to provide well-informed estimations of DNI.

Numerical methods employ numerical weather prediction (NWP) models [24–26] to estimate cloud cover position and extent [27]. These models can be global, such as the NOAA/NCEP Global Forecast System (GFS) [28], or a combination of global and local methods, like forecasts derived from the National Digital Forecast Database (NDFD) [29]. NWP models are typically used for prediction horizons of one or more days ahead [30], while data-based models demonstrate higher accuracy for short-term forecast horizons that typically ranging from 3 to 5 h [27,30]. These models derive cloud cover estimates from historical data obtained through satellite-based remote sensing or ground-based instrumentation, or a combination of the two [20,31–36].

Satellite-based DNI forecasts currently suffer from low temporal and spatial resolution. To address this limitation, models using digital processing of ground-based sky imagery are gaining popularity [37–42]. However, implementing these ground-based sensors at scale can be costly. While efforts have been made to develop cost-effective sensor networks [39,43–47], the collection of high-quality meteorological data still remains a challenge, especially in the developing world. This is particularly concerning as many developing countries are located near the equator and could greatly benefit from concentrated solar power technology to address their energy challenges.

Data-driven approaches using machine learning and advanced deep neural networks have also been explored in the literature. These studies typically formulate the DNI forecasting problem as a time-series regression task [20–22]. While these approaches have shown that higher accuracies can be achieved using data-driven methods, they rely on meteorological parameters as features for the models. As mentioned earlier, acquiring high-quality meteorological data is a challenge in itself. We formulate DNI forecasting as a classification problem, which to the best of our knowledge, has never been previously undertaken. This study aims to develop DNI forecast models that capture the implicit dependence of DNI on *time*, thus requiring no meteorological data. This task represents a substantially greater challenge compared to modeling the explicit dependence based on meteorological factors. Furthermore, this work offers an exceptional opportunity to demonstrate the immense potential of modern deep learning methods, such as Residual Networks [48], Bidirectional LSTMs [49,50], and Transformers [51], in the field of DNI modeling. Despite the remarkable performance demonstrated by these methods in complex sequence modeling tasks [52–58] their application in the domain of DNI forecasting has received limited attention.

3. Methodology

In this section, we introduce our classification-based formulation of the problem of DNI forecasting in the absence of meteorological data. We present four modern neural classifiers specifically designed to address this problem, which include a Deep ReLU Multilayer Perceptron, a 1-dimensional Residual Neural Network, a Multilayer Bidirectional LSTM, and a Transformer.[1] These neural network models are inherently adaptable. Therefore, if meteorological data is available for a region, extending these models to incorporate and leverage this data, either through pre-training [61] on such data or though knowledge distillation [62] from another network trained on it, could be a straightforward endeavor.

3.1. Problem formulation

We frame the problem of DNI modeling as a time-series classification task, in which the objective is to predict whether the DNI of the sun would fall within pre-defined limits at a given time. Our approach is motivated by the fact that in most DNI forecasting applications, it is sufficient to know whether the DNI is within certain bounds rather than the exact value. For example, in a CSP plant with thermal energy storage, there is a DNI threshold for which the plant's collection system can either start up the receiver to collect heat for recirculation back to

[1] We refer the reader to Goodfellow et al. [59] and Zhang et al. [60] for an introduction to these deep neural network architectures.

Table 1
Tabular visualization of the dataset $\mathcal{D}$ [63]. Each row i in the table corresponds to a pair $(t_i, y_i) \in \mathcal{D}$, where $t_i = (\text{Year}, \text{Month}, \text{Day}, \text{Hour}, \text{Minute})_i \in \mathbb{R}_+^D$ is a D-dimensional vector, with $D = 5$, defining the time at which the corresponding measurement of direct normal irradiance $y_i = \text{DNI}_i \in \mathbb{R}_+$ was made.

	Year	Month	Day	Hour	Minute	DNI
1	1998	01	01	00	30	0
2	1998	01	01	01	30	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
9	1998	01	01	08	30	46
10	1998	01	01	09	30	0
11	1998	01	01	10	30	355
12	1998	01	01	11	30	248
⋮	⋮	⋮	⋮	⋮	⋮	⋮
201478	2020	12	31	21	30	0
201479	2020	12	30	22	30	0
201480	2020	12	31	23	30	0

the cold salt tank, and a second threshold at which there is sufficient DNI to pump salt through the receiver to the hot salt tank for electricity generation [15], leading to three classes for operation; a fourth class might be used to determine when to defocus some heliostats in the solar field to prevent damage to the receiver due to excessive flux [12].

The classification framework is also particularly suitable considering the complexity of DNI modeling in the absence of meteorological data, where treating the problem as a regression task would yield less informative results. Furthermore, a regression treatment would yield inaccurate models due to the noise in DNI measurements. Hence, we formulate the DNI modeling task as to approximate the function $g : \mathbb{R}_+^D \to \{0, 1, \dots, C-1\}$ defined as:

$$g(t) = \begin{cases} 0 & \text{if } \delta_0 \le f(t) < \delta_1, \\ 1 & \text{if } \delta_1 \le f(t) < \delta_2, \\ \vdots & \\ C-1 & \text{if } \delta_{C-1} \le f(t) < \delta_C, \end{cases}$$

where $f : \mathbb{R}_+^D \to \mathbb{R}_+$ represents the implicit dependence of DNI on time, and the thresholds $0 \le \delta_0 < \delta_1 < \cdots < \delta_C$ discretize the range of f into C classes. Since f is unknown, the aim is to construct the approximation to g from noisy snapshots of f in the form of a coarse dataset:

$$\mathcal{D} = \left\langle (t_1, y_1), \dots, (t_N, y_N) \text{ s.t. } t_i \in \mathbb{R}_+^D, y_i \in \mathbb{R}_+ \right\rangle,$$

consisting of DNI observations made at 1 hour intervals, such that each observation $(t_i, y_i) \in \mathcal{D}$ is a sampling of f under additive noise ϵ_i, i.e., $y_i = f(t_i) + \epsilon_i$. Note that due to the noise ϵ_i, the true class of a DNI value y_i cannot be known. Instead, we assign a DNI value y_i to a class $C(y_i)$ if $\delta_{C(y_i)} \le y_i < \delta_{C(y_i)+1}$.[2]

3.2. Data processing

Table 1 provides an excerpt of the dataset $\mathcal{D}$ of 201,480 observations used in this paper. The data has been obtained from the National Solar Radiation Database (NSRDB) [63] for the location of Tonopah, NV, USA. Tonopah was chosen due to its consistent solar resource availability throughout the year and its existing CSP project.

Following the recommended practice in data science [64], we randomly split the dataset $\mathcal{D}$ into three subsets: $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{val}}$, and $\mathcal{D}_{\text{test}}$, where $|\mathcal{D}_{\text{train}}| = 141{,}036$ and $|\mathcal{D}_{\text{val}}| = |\mathcal{D}_{\text{test}}| = 30{,}222$. The proposed models are trained using $\mathcal{D}_{\text{train}}$, and their performance on $\mathcal{D}_{\text{val}}$ is used as a criterion for saving the models during training epochs. While both $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$ are utilized in training the models, $\mathcal{D}_{\text{test}}$ is kept separate to provide an unbiased assessment of performance once training has been completed.

[2] An alternative approach might be to assign y_i to $C(y_i)$ if $\delta_{C(y_i)} \le y_i - \epsilon_i < \delta_{C(y_i)+1}$, where $\epsilon_i \sim \mathcal{N}(\mu_\epsilon, \sigma_\epsilon^2)$, for some μ_ϵ and σ_ϵ. However, this is left for future research.

3.2.1. Feature selection

We used the Python library Pandas [65,66] to identify redundant features in the dataset. By eliminating these redundant features, we effectively reduced the dimensionality of the feature space to $d = 4 < D$.

To gain deeper insights into the influence of various features on the generalizability of machine learning models, we employed the Matplotlib library [67] to plot and analyze the average variation in DNI over different timescales, as shown in Figs. 1(a) –1(b). The plots reveal that DNI exhibits regular variations on monthly and daily timescales, which correspond to the seasonal and diurnal nature of the resource, respectively. Additionally, the plots show a subtle variation in DNI from year to year, which encodes larger-scale weather phenomena, such as global warming/cooling trends and their effect on the variability of the weather.

3.2.2. Z-score normalization and min–max scaling

As the final preprocessing step, we normalized the feature space to zero-center the input distribution and make its spread uniform across all feature dimensions [64]:

$$\tilde{t}_i^{(j)} = \frac{t_i^{(j)} - \mu^{(j)}}{\sigma^{(j)}}, \quad \forall (t_i, y_i) \in \mathcal{D},$$

where $t_i^{(j)}$ represents the j-the component of the ith input feature, and

$$\mu^{(j)} = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(t,y)\in\mathcal{D}_{\text{train}}} t^{(j)}, \quad \sigma^{(j)} = \sqrt{\frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(t,y)\in\mathcal{D}_{\text{train}}} \left(t^{(j)} - \mu^{(j)}\right)^2}.$$

Furthermore, we applied min–max normalization to the outputs to map them onto the interval $[0, 1]$:

$$\tilde{y}_i = \frac{y_i - \min_{(t,y)\in\mathcal{D}_{\text{train}}} y}{\max_{(t,y)\in\mathcal{D}_{\text{train}}} y - \min_{(t,y)\in\mathcal{D}_{\text{train}}} y},$$

for all $(t_i, y_i) \in \mathcal{D}$.

3.3. Deep learning models

We developed multiple deep neural networks for approximating the DNI function g from $\mathcal{D}_{\text{train}}$. Each neural network produces a C-dimensional vector, representing a probability distribution over the C DNI classes. Hence, for a given neural network Γ, we can construct a model of g as $\hat{g} = \text{argmax} \circ \Gamma$.

To train the proposed deep neural networks so that $\hat{g}$ approximates g accurately, we use a computationally efficient variant of stochastic gradient descent called *Adam* [68], and we design a novel schedule that causes the rate of descent of the algorithm, α, to incur attenuated exponential decay after k_0 epochs:

$$\alpha_k = \begin{cases} \alpha_{k-1}, & \text{if } k < k_0, \\ \alpha_{k-1}\left(1 - \tau/\xi(k)\right) & \text{otherwise}, \end{cases}$$

in which τ is the decay rate, and $\xi(k) = e^{\nu(k-k_0)}$ attenuates the decay according to some given attenuation factor ν. Fig. 2 shows our novel learning rate decay schedule (un-marked solid line) attenuates the rate of decay towards the later epochs. In comparison, previous exponential decay schedules (marked line and dashed line) nearly zero out the learning rate towards the later epochs, thus hampering learning. The optimizer's objective is to find a parameterization of Γ that minimizes the in-sample stochastic cross-entropy loss:

$$\mathcal{L}(\Gamma) = -\frac{1}{|\mathcal{B}|} \sum_{(\tilde{t}_i, y_i)\in\mathcal{B}} \sum_{c=0}^{C-1} \mathbb{1}\left[c = C(y_i)\right] \log \hat{p}_i^{(c)}(\Gamma) + \mathcal{R}(\Gamma).$$

Here, $\hat{p}_i(\Gamma) = \Gamma(\tilde{t}_i)$ if Γ is the Multilayer Perceptron proposed in Section 3.3.1, and

$$\hat{p}_i(\Gamma) = \Gamma\left(\langle \tilde{t}_{i-k}, \tilde{y}_{i-k} \rangle_{k=1}^K\right)$$

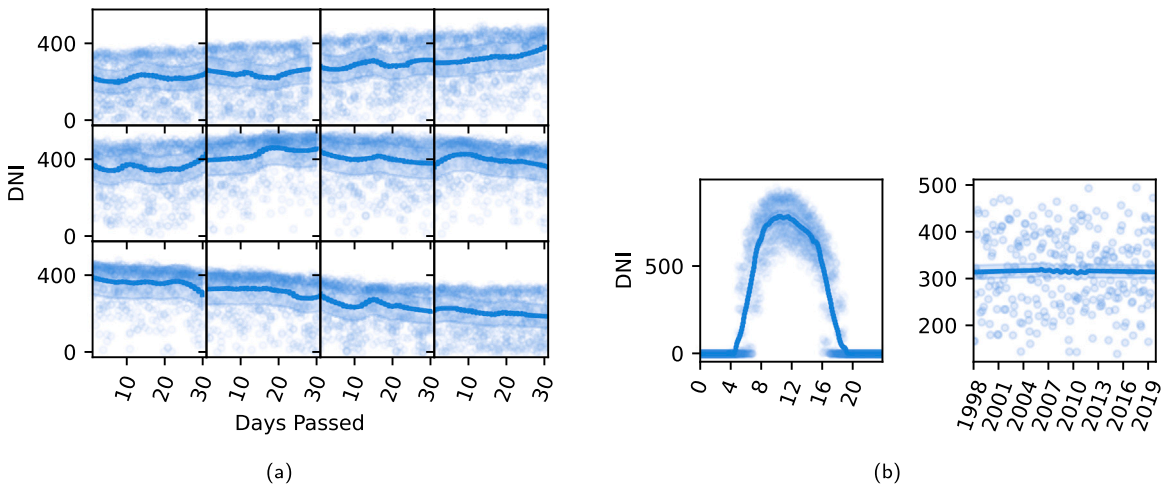

Fig. 1. (a) Average DNI variation over the days of a year. From left to right, the top row corresponds to Jan–Apr, the middle row to May–Aug, and the bottom row to Sep–Dec. (b). The left plot shows the average DNI each passing hour of the day, and the right plot shows the average DNI over the years 1998–2020.

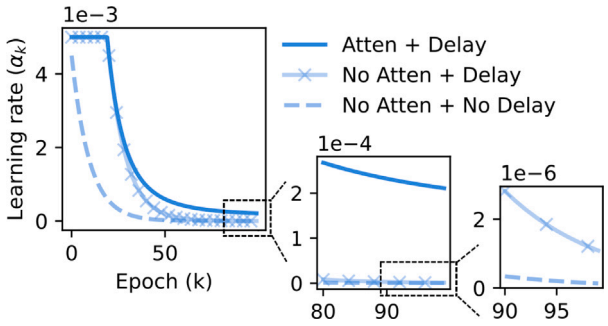

Fig. 2. Learning rate decay schedule.

if Γ is one of the sequence-processing architectures presented in Sections 3.3.2–3.3.4. Furthermore, the set $\mathcal{B} \subset \mathcal{D}_{\text{train}}$ represents a batch of training samples, and the term $\mathcal{R}(\Gamma)$, called *regularization*, incentivizes lower-order parameterization of Γ to limit overfitting. For the proposed neural networks, we regularize only the affine parameters.

3.3.1. Deep multilayer perceptron (MLP)

The first neural architecture we propose is a multilayer perceptron, as shown in Fig. 3A, where each blue block represents a fully-connected layer. The properties of the layer include (in order) (1) the number of neurons (n_ℓ), (2) activation type (Ψ), (3) presence of batch-normalization, and (4) dropout percentage (p_ℓ). MLP $\Gamma : \mathbb{R}^d \to [0,1]^C$ of the form:

$$\Gamma(\tilde{t}) = \sigma\left(A_L \Phi\left(\dots\left(A_2 \Phi\left(A_1 \tilde{t} + b_1\right) + b_2\right)\dots\right) + b_L\right).$$

Here,

$$\sigma^{(j)}(u_L) = \frac{1}{\sum_{c=1}^{C} \exp u_L^{(c)}} \exp u_L^{(j)}$$

is the softmax function, $L \in \mathbb{N}$ is the number of layers in the network, and the parameters $A_\ell \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$, $b_\ell \in \mathbb{R}^{n_\ell}$, for $\ell \in \{1, \dots, L\}$, define the size of each layer, where $n_0 = d$ and $n_L = C$. We set $L = 7$, and we define the mapping Φ as the composition of batch normalization, followed by a ReLU activation Ψ, and dropout:

$$\Phi(u_\ell) = \eta_\ell \odot \Psi\left(\text{BN}_{\gamma_\ell, \beta_\ell}\left(u_\ell\right)\right),$$

in which $\gamma_\ell, \beta_\ell \in \mathbb{R}^{n_\ell}$ are learnable parameters of the batch normalization layer, and $\mathbb{R}^{n_\ell} \ni \eta_\ell \sim \text{Bern}\left(p_\ell\right)$ is a vector of independent Bernoulli random variables. Batch normalization (BN) stabilizes the training of the deep network by re-centering and re-scaling the affine output $u_\ell = A_\ell \Phi u_{\ell-1} + b_\ell$ (where $\Phi(u_0) = t$) of each layer [69]. The ReLU activation, $\Psi^{(j)}(x) = (x^{(j)})_+ = \max(x^{(j)}, 0)$, adds computationally efficient nonlinear transformations to the network [70], and dropout zeros out these activations with some probability p_ℓ to "motivate" the network to learn distributed activation patterns, thus alleviating overfitting [71].

3.3.2. 1D-ResNet

We introduce a modified version of the widely adopted ResNet architecture [48] which we tailor specifically to classify DNI. The base unit of our ResNet variant is a *1D Residual Block*, defined as:

$$\text{Res}(x) = \text{Conv}_\ell^2\left(\text{Conv}_\ell^1(x)\right) + \text{Conv}_\ell^3(x).$$

Here, Conv_ℓ represents a 1-dimensional convolutional operation with a given set of filters, optionally followed by batch normalization and ReLU. The complete 1D-ResNet architecture, shown in Fig. 3B, comprises eight such residual blocks. The yellow blocks are 1D convolution operations, and their properties include (in order): (1) number of filters, filter size, and stride length; (2) activation type, (3) presence of batch-normalization, and (4) dropout percentage. The input to the network is a sequence of K pairs, $\langle(\tilde{t}_{i-k}, \tilde{y}_{i-k})\rangle_{k=1}^{K}$, which represent the history of DNI up to time t_{i-1}, and the network predicts the class to which the DNI at time t_i belongs. By leveraging DNI history and incorporating efficient architectural features such as skip-connections for effective training, and weight-sharing for translation invariance, our 1D-ResNet is able to generally output more accurate predictions than the MLP model proposed in Section 3.3.1.

3.3.3. Multilayer bidirectional LSTM

Long short-term memory (LSTM) neural networks are a type of recurrent neural networks (RNNs) designed to capture long-term dependencies in sequences more effectively [50]. Similar to vanilla recurrent networks, LSTMs employ a recursive unit ζ to maintain a hidden state $h_j \in \mathbb{R}^w$ as they sequentially process the elements $(\tilde{t}_j, \tilde{y}_j)$ of the sequence $\langle(\tilde{t}_{i-k}, \tilde{y}_{i-k})\rangle_{k=1}^{K}$. However, unlike vanilla recurrent networks, LSTM cells incorporate a sophisticated gating mechanism to control the updates to their hidden state:

$$h_j = \zeta\left(t_j, h_{j-1}\right) = \tanh c_j \odot \sigma\left(A_{ot} t_j + A_{oh} h_{j-1} + b_o\right),$$

where $c_j = g_j \odot i_j + c_{j-1} \odot f_j$, called the *cell state*, is defined in terms of the gates

$$g_j = \tanh\left(A_{gt} t_j + A_{gh} h_{j-1} + b_g\right) \qquad \textit{Cell Gate}$$

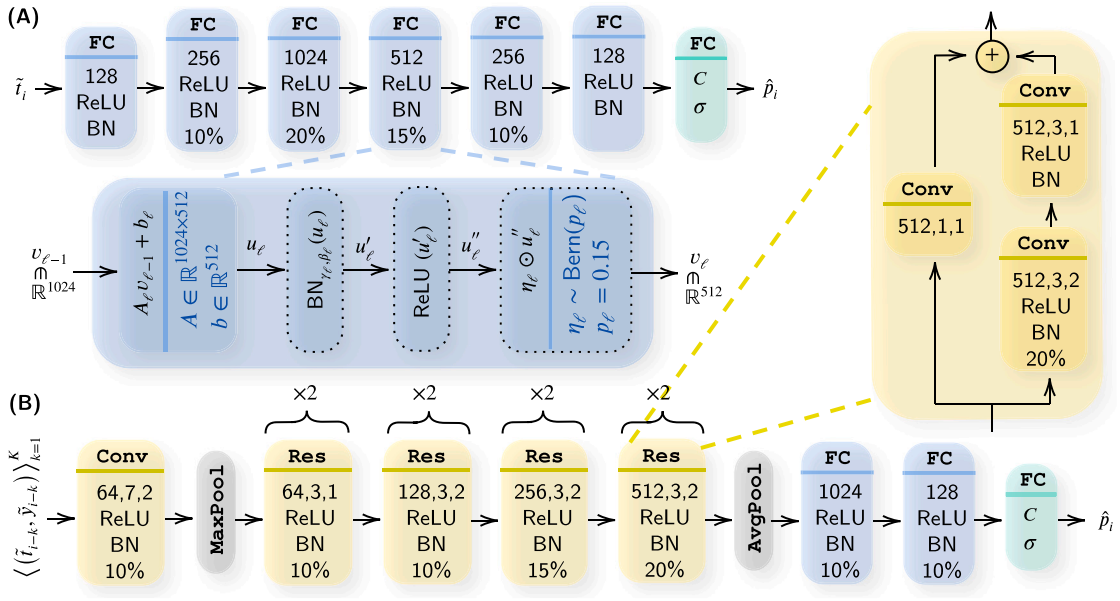

Fig. 3. **(A)** Deep Multilayer Perceptron architecture proposed in Section 3.3.1. **(B)** Architecture of the 1D-ResNet from Section 3.3.2. Note that the final layer of each network is a fully-connected layer with softmax activation, yielding a probability distribution $\hat{p}_i$ over the C DNI categories.

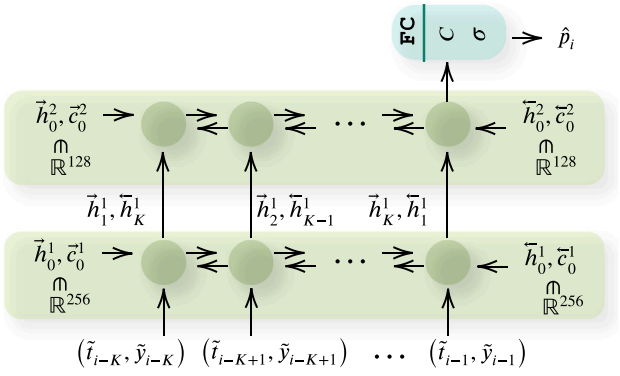

Fig. 4. A 2-layer bidirectional LSTM for DNI classification shown in its unrolled form. Each green circle represents a Bidirectional LSTM unit with recurrent dropout and input dropout percentages of 5%.

$$i_j = \sigma\left(A_{it}t_j + A_{ih}h_{j-1} + b_i\right) \qquad \textit{Input Gate}$$
$$f_j = \sigma\left(A_{ft}t_j + A_{fh}h_{j-1} + b_f\right) \qquad \textit{Forget Gate}$$

Here, $A_{ot}, A_{gt}, A_{it}, A_{ft} \in \mathbb{R}^{w\times d}$, $A_{oh}, A_{gh}, A_{ih}, A_{fh} \in \mathbb{R}^{w\times w}$ and $b_o, b_g, b_i, b_f \in \mathbb{R}^w$ are trainable parameters of ζ.

For DNI estimation, we propose a recurrent architecture based on bidirectional LSTMs (BiLSTMs). Unlike traditional LSTMs, BiLSTMs maintain two hidden states: $\overrightarrow{h}_j \in \mathbb{R}^w$, encoding the historical context at each step j, and $\overleftarrow{h}_j \in \mathbb{R}^w$, encoding future context. Moreover, in our architecture, we stack two BiLSTM layers, as depicted in Fig. 4, where the lower layer has a width (w) of 256 neurons and the upper layer has a width of 128 neurons. The stacking enables the network to achieve spatial depth, which allows for both low-order and high-order representations of the forward and backward context to be encoded at each step j.

3.3.4. Transformer

Similar to recurrent networks, transformers are neural networks designed to process sequential data. However, instead of processing a sequence one element at a time, transformers encode the whole sequence as a matrix X and employ a mechanism called *attention* [51] to attend to the entirety of X at once.

To forecast DNI, we develop a transformer model that utilizes an embedding layer to encode the positions $1, \ldots, K$ of the sequence elements $(\tilde{t}_{i-K}, \tilde{y}_{i-K}), \ldots, (\tilde{t}_{i-1}, \tilde{y}_{i-1})$ as dense vectors $p_1, \ldots, p_K \in \mathbb{R}^5$. Each positional encoding p_j is then added to its corresponding sequence element $(\tilde{t}_j, \tilde{y}_j)$, and the resulting vectors $x_j \in \mathbb{R}^5$ are concatenated to form the matrix $X \in \mathbb{R}^{K\times 5}$:

$$x_j = \begin{bmatrix} x_j^{(1)} \\ x_j^{(2)} \\ x_j^{(3)} \\ x_j^{(4)} \\ x_j^{(5)} \end{bmatrix} = p_j + \begin{bmatrix} \tilde{t}_j^{(1)} \\ \tilde{t}_j^{(2)} \\ \tilde{t}_j^{(3)} \\ \tilde{t}_j^{(4)} \\ \tilde{y}_j^{(5)} \end{bmatrix}, \quad X = \begin{bmatrix} — & x_1^T & — \\ — & x_2^T & — \\ — & x_3^T & — \\ & \vdots & \\ — & x_K^T & — \end{bmatrix}$$

This new representation of the input sequence, X, is subsequently processed by two back-to-back transformer blocks. Having two transformer blocks facilitates the encoding of both low-order and high-order representations of X. Moreover, to enable the network to attend to different parts of these representations differently, we equip each transformer block with multi-headed attention units. The complete model, depicted in Fig. 5, achieves prediction accuracy comparable to that of the 2-layer BiLSTM model at a significantly reduced prediction time. This makes the transformer architecture particularly suitable for real-time or near-real-time use cases of DNI estimation.

4. Results

Table 2 presents the performance of the proposed models on various instances of the DNI classification problem. These instances have been constructed by varying the number of classes C from 2 to 10, such that we allocate class 0 to represent DNI values of 0, while classes 1 to $C-1$ are assigned in a manner that aims for approximately equal distribution of DNI values among these classes. For example, a three-class scenario can be used for CSP plants, in which the three classes (DNI bounds) identify whether the thermal flux is (i) not sufficient to operate the plant, (ii) sufficient to recirculate salt to the cold tank after collection, or (iii) sufficient to collect heat for transfer to the hot salt tank. We provide a visual representation of Table 2 in Fig. 6, where a prediction is considered accurate if the predicted DNI class matches the actual

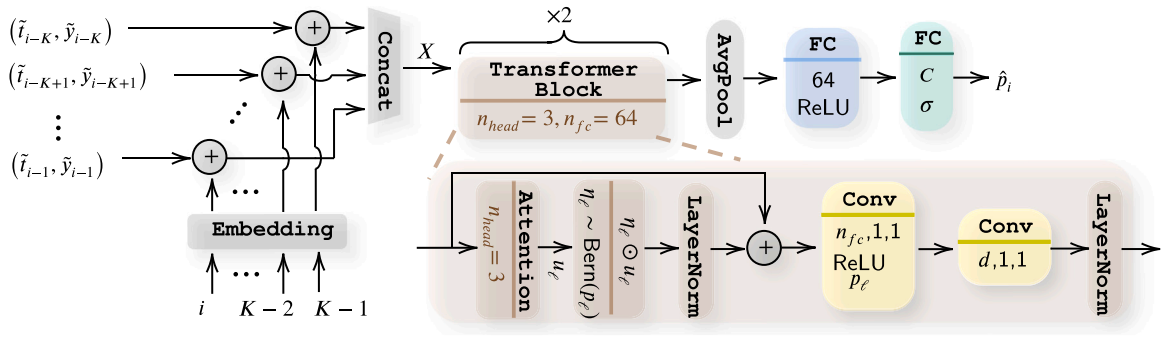

Fig. 5. A powerful but minimal transformer model for highly accurate and near real-time DNI estimation. The model incorporates each input's position into its embedding and processes it through two adjacent transformer blocks.

Table 2
Quantitative performance of the proposed models. The "Train", "Val", and "Test" columns give each model's percentage of correct classifications on D_{train}, D_{val}, and D_{test}, respectively. The "Run" column identifies the experiments we ran for each model and number of classes.

C	ReLU-MLP			1D-ResNet			BiLSTM			Transformer		
	Train	Val	Test	Train	Val	Test	Train	Val	Test	Train	Val	Test
2	87.82	88.03	87.75	93.77	92.57	92.88	94.29	94.28	93.42	94.33	93.88	93.07
3	85.57	85.93	85.39	92.25	85.26	88.32	92.74	92.04	92.01	93.27	91.60	92.19
4	81.00	81.28	80.89	86.99	79.29	83.46	90.27	87.33	89.19	89.47	87.32	87.92
5	77.63	77.93	77.57	83.34	76.90	79.09	87.78	84.70	86.29	87.27	84.23	85.06
6	74.47	74.63	74.44	80.06	72.18	76.41	86.41	82.55	84.55	85.66	81.65	82.98
7	72.92	72.53	72.43	79.48	71.33	76.10	84.48	79.42	82.60	83.32	77.07	80.56
8	70.59	70.49	70.35	79.71	91.50	74.58	83.98	78.17	81.52	82.10	76.23	79.06
9	69.27	69.39	69.27	75.75	66.17	69.95	83.24	76.83	80.29	81.32	75.61	78.53
10	67.50	67.45	67.72	75.49	65.67	69.99	81.65	75.32	78.91	81.55	74.71	78.17

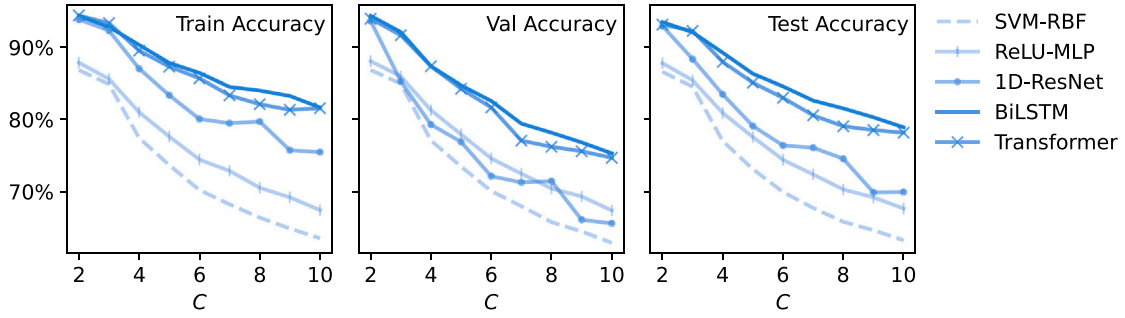

Fig. 6. From left to right: accuracy of the proposed deep neural network models on D_{train}, D_{val} and D_{test}, respectively, in terms of C, the number of classes. For comparison, results for a support vector machine with radial basis function kernel (SVM-RBF) have also been provided.

DNI class. Additionally, for the purpose of comparison with traditional machine learning techniques, the figure includes the performance of a Support Vector Machine with Radial Basis Function kernel (SVM-RBF). The hyperparameters for all the models were selected using an automatic routine called Grid Search [72]. Grid Search involves specifying a range of possible values for each hyperparameter and then systematically evaluating the model performance for every possible combination of these values. Grid Search is widely used in machine learning and is supported by many machine learning libraries, including Keras and Scipy, used in this study.

The information conveyed by Fig. 6 clearly illustrates that our proposed deep learning models consistently outperform SVM-RBF. Our BiLSTM model demonstrates the highest accuracy among all deep learning models, with the transformer model exhibiting slightly lower accuracy. However, the additional insights in the results in Fig. 7 and Table 3 show that the transformer model is significantly more efficient in computation and memory when compared to the BiLSTM model. Following the usual practice in machine learning, we measure computational complexity in terms of the total number of floating point operations evaluated in one forward pass of a network. In addition, we also measure the inference time of the models in seconds, which provides a more practical criterion for evaluating time-efficiency. Regarding memory-efficiency, we simply take the number of trainable parameters composing the model as our criterion, since all the parameters are stored as floating point values of the same precision. As evident from 7 and Table 3, the transformer model's efficiency positions it as the more appropriate choice for deployment in scenarios with limited computational resources, such as real-time control applications running on resource-constrained platforms like microcontrollers. It is also evident from a comparison of the accuracy and efficiency numbers that the BiLSTM model, though most accurate, is the least efficient, with long computation times and a large memory footprint.

Our experiments further shed light on the performance of the ReLU-based Multilayer Perception presented in Section 3.3.1, which fares the worst among all the proposed deep learning models. The 1D-ResNet introduced in Section 3.3.2, although superior to the ReLU-MLP in terms of performance, underperforms relative to the other deep learning models. We believe that the input's dimensionality may not be high enough to effectively leverage the spatial depth of the 1D-ResNet. However, in a transfer learning scenario where the models are pretrained with access

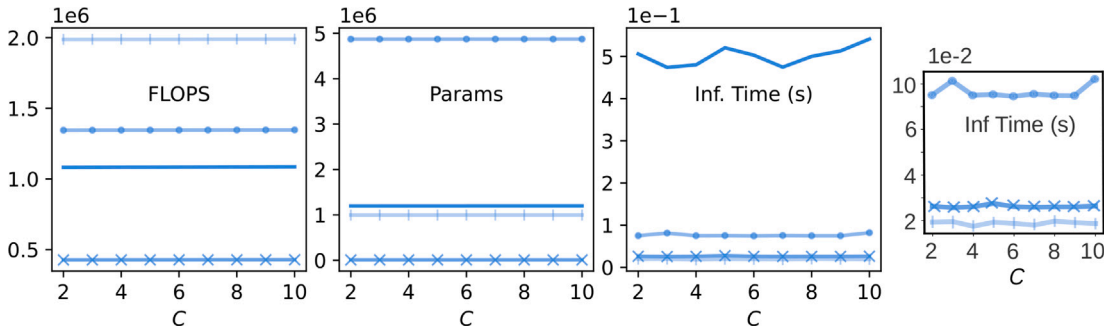

Fig. 7. The time and space footprint of the proposed models, as a function of C, the number of classes. From left to right: the first plot provides the number of floating point operations (FLOPs) in the models; the second is the number of trainable parameters; and the last two plots give the inference time (inf. Time) of the models. (The legend is in Fig. 6).

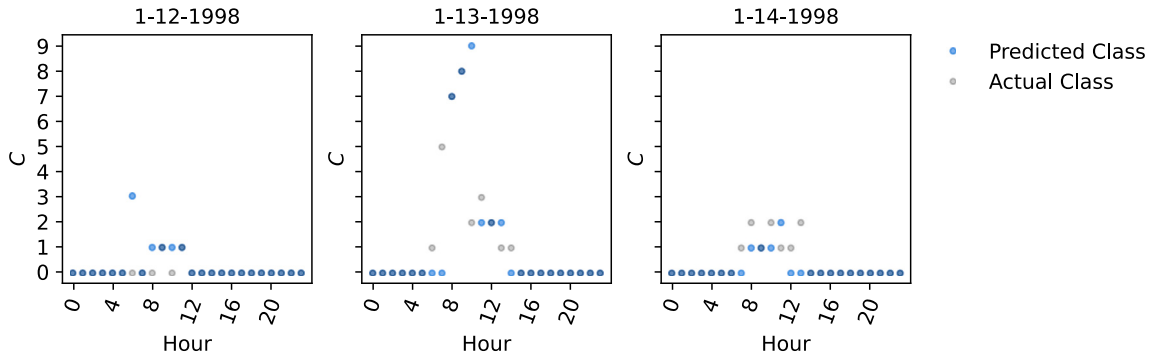

Fig. 8. Comparison of predictions from the 10-class transformer model to the actual DNI classes over a period of 3 consecutive days randomly chosen from the test dataset.

Table 3
Average test accuracy, parameter count, floating-point operations (FLOPS), and inference time (in milliseconds) for the proposed deep neural networks, computed across the number of classes (C).

	ReLU-MLP	1D-ResNet	BiLSTM	Transformer
Test Accuracy	76.20%	78.98%	85.42%	84.17%
Parameters	995,855	4,872,147	1,196,563	8215
FLOPS	1,989,424	1,346,090	1,083,482	427,114
Inference Time	18.91	76.51	501.44	26.24

to meteorological data, the 1D-ResNet might extract more meaningful features and potentially outperform the comparatively spatially shallow BiLSTM.

An important behavior depicted in Fig. 7 is that the complexity of our models shows minimal change as the problem complexity (i.e., C) increases. Fig. 6 shows that the accuracy of the models decreases considerably with increase in C. A promising avenue for future research is to develop dynamic models that adapt (under a hand-coded or learned policy) as C grows, enabling more smoother performance scaling for complex DNI estimation scenarios. An error analysis for the forecasted DNI is shown in Fig. 8 where we compare the predicted and actual DNI thresholds (classes) from the testing data. It can be seen that the model performs well even on overcast days (1-12-1998 and 1-14-1998), suggesting that the model has implicitly learned a weather model. Furthermore, when the model's prediction fails, it often falls short by only one class margin, demonstrating the robustness of the model.

5. Conclusion

We present a framework for DNI forecasting using only time as input instead of meteorological data. Specifically, we introduce a novel multi-class classification formulation of the DNI estimation problem that is based on the implicit dependence of DNI on time rather than meteorological parameters, and we propose four cutting-edge neural networks to model it. Our experimental results reveal that the BiLSTM achieves the highest accuracy among the proposed deep models, making it ideal for offline DNI estimation scenarios. In contrast, for online applications in which fast inference speed and low memory footprint are important, our transformer model emerges as a compelling choice, offering competitive accuracy at significantly lower computational expense. Additionally, we propose a ResNet variant for DNI estimation, which we posit could be particularly effective for high-dimensional DNI estimation problems.

By enabling DNI estimation without the need for meteorological data, which can be difficult and expensive to obtain, we aim to pave the way for innovative approaches to optimizing forecast-informed CSP operations and contribute to global efforts in combating climate change. Furthermore, we hope to inspire fellow researchers to explore the integration of modern AI techniques into DNI estimation. Throughout this paper, we identify several promising avenues for future research in this direction, including the incorporation of noise prior, leveraging transfer learning and knowledge distillation to integrate meteorological data into the models, and adapting network sizes in proportion to the number of classes.

The location selected for this study, Tonopah, NV, has been the subject of many CSP studies due to its high DNI resource and significance as a CSP site. By focusing on this well-studied location, we are able to set a baseline for evaluating our novel classification approach. Our goal in this study was to establish the effectiveness of the classification framework in a controlled environment with high-quality data before exploring broader applications. Another intriguing direction for future work could involve utilizing latitude and longitude values as inputs to the network. This would include developing and testing multi-location models that account for varying weather patterns and DNI resources, and utilizing them to predict DNI values at intermediate locations. Applying the models to these additional locations would provide further evidence of their generalizability. In this work, we use hourly data;

future studies can consider higher temporal fidelity to provide fine-grained DNI estimates. We postulate that the models proposed in this paper can be used ad verbatim for fine-grained predictions through training on higher fidelity data. The application of modern AI has transformed numerous engineering fields, and we hope that our work demonstrates that CSP technology may benefit significantly from such advancements.

CRediT authorship contribution statement

Muhammad Saud Ul Hassan: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Kashif Liaqat:** Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Laura Schaefer:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Alexander J. Zolan:** Writing – review & editing, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to acknowledge the *Ken Kennedy Institute 2023/24 Scott Morton Memorial (provided by the Energy HPC Conference) Graduate Fellowship*.

This work was co-authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. 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 work, or allow others to do so, for U.S. Government purposes.

Data availability

The dataset used in this work was obtained from the National Solar Radiation Database (NSRDB) and is available publicly.

References

[1] Hannah Ritchie, Max Roser, Pablo Rosado, Energy, Our World Data (2022) https://ourworldindata.org/energy.

[2] IA Bashmakov, LJ Nilsson, A Acquaye, C Bataille, JM Cullen, M Fischedick, Y Geng, K Tanaka, et al., Climate change 2022: Mitigation of climate change. Contribution of working group III to the sixth assessment report of the intergovernmental panel on climate change, chapter 11, 2022.

[3] Intergovernmental Panel on Climate Change, Global Warming of 1.5° C: an IPCC Special Report on the Impacts of Global Warming of 1.5° C Above Pre-Industrial Levels and Related Global Greenhouse Gas Emission Pathways, in the Context of Strengthening the Global Response to the Threat of Climate Change, Sustainable Development, and Efforts to Eradicate Poverty, Intergovernmental Panel on Climate Change, 2018.

[4] Fortune Business Insights, Concentrated Solar Power (CSP) Market Size, Share & COVID-19 Impact Analysis, By Technology (Parabolic Trough, Power Tower, Linear Fresnel), By Application (Residential, Non-Residential, Utility) and Regional Forecasts, 2021–2028, 2021, Markey Research Report. Nov, 2021.

[5] R.P. Merchán, M.J. Santos, A. Medina, A. Calvo Hernández, High-temperature central tower plants for concentrated solar power: 2021 overview, Renew. Sustain. Energy Rev. 155 (2022) 111828.

[6] Kashif Liaqat, Juan C. Ordonez, Laura Schaefer, Alexander J. Zolan, Design and techno-economic analysis of a 150-MW hybrid CSP-PV plant, in: 2023 IEEE Conference on Technologies for Sustainability, SusTech, IEEE, 2023.

[7] Kashif Liaqat, Juan C. Ordonez, Design and optimization of CSP power plants for Pakistan: a comparative study, Clean Energy 7 (3) (2023) 690–704.

[8] P. Blanc, B. Espinar, N. Geuder, C. Gueymard, R. Meyer, R. Pitz-Paal, B. Reinhardt, D. Renné, M. Sengupta, L. Wald, S. Wilbert, Direct normal irradiance related definitions and applications: The circumsolar issue, Sol. Energy 110 (2014) 561–577.

[9] Edward W Law, Abhnil A Prasad, Merlinde Kay, Robert A Taylor, Direct normal irradiance forecasting and its application to concentrated solar thermal output forecasting–A review, Sol. Energy 108 (2014) 287–307.

[10] Nate Blair, Aron P Dobos, Janine Freeman, Ty Neises, Michael Wagner, Tom Ferguson, Paul Gilman, Steven Janzou, System advisor model, sam 2014.1. 14: General description, 2014.

[11] Michael J. Wagner, Tim Wendelin, SolarPILOT: A power tower solar field layout and characterization tool, Sol. Energy 171 (2018) 185–196.

[12] Alexander Zolan, William Hamilton, Michael Wagner, Kashif Liaqat, Solar Field Layout and Aimpoint Strategy Optimization, National Renewable Energy Laboratory (NREL), Golden, CO (United States), 2021.

[13] Alexander Zolan, William Hamilton, Kashif Liaqat, Michael Wagner, USDOE Office of Energy Efficiency, Renewable Energy, HALOS (heliostat aimpoint and layout optimization software), 2021, http://dx.doi.org/10.11578/dc.20210616.1.

[14] Lorin L. Vant-Hull, The Role of "Allowable Flux Density" in the Design and Operation of Molten-Salt Solar Central Receivers , J. Sol. Energy Eng. 124 (2) (2002) 165–169.

[15] John L Cox, William T Hamilton, Alexandra M Newman, Michael J Wagner, Alex J Zolan, Real-time dispatch optimization for concentrating solar power with thermal energy storage, Optim. Eng. 24 (2) (2023) 847–884.

[16] Gökçe Kahvecioğlu, David P Morton, Michael J Wagner, Dispatch optimization of a concentrating solar power system under uncertain solar irradiance and energy prices, Appl. Energy 326 (2022) 119978.

[17] Arsalan Mousavian, Dragomir Anguelov, John Flynn, Jana Kosecka, 3D bounding box estimation using deep learning and geometry, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7074–7082.

[18] Raied Salman, Vojislav Kecman, Regression as classification, in: 2012 Proceedings of IEEE Southeastcon, 2012, pp. 1–6.

[19] Sholom M. Weiss, Nitin Indurkhya, Rule-based machine learning methods for functional prediction, J. Artificial Intelligence Res. 3 (1995) 383–403.

[20] Youssef Karout, Stéphane Thil, Julien Eynard, Emmanuel Guillot, Stéphane Grieu, Hybrid intrahour DNI forecast model based on DNI measurements and sky-imaging data, Sol. Energy 249 (2023) 541–558.

[21] Chengying Zhang, Zhen Yan, Chao Ma, Xiaoli Xu, Prediction of direct normal irradiance based on ensemble deep learning models, in: 2020 IEEE 3rd International Conference on Electronics Technology, ICET, 2020, pp. 425–432.

[22] Felipe P. Marinho, Paulo A.C. Rocha, Ajalmar R.R. Neto, Francisco D.V. Bezerra, Short-Term Solar Irradiance Forecasting Using CNN-1D, LSTM, and CNN-LSTM Deep Neural Networks: A Case Study With the Folsom (USA) Dataset, J. Sol. Energy Eng. 145 (4) (2022) 041002.

[23] Dorota Matuszko, Influence of the extent and genera of cloud cover on solar radiation intensity, Int. J. Climatol. 32 (15) (2012) 2403–2414.

[24] Elke Lorenz, Jan Remund, Stefan C Müller, Wolfgang Traunmüller, Gerald Steinmaurer, David Pozo, José Antonio Ruiz-Arias, V Lara Fanego, Lourdes Ramirez, Martin Gaston Romeo, et al., Benchmarking of different approaches to forecast solar irradiance, in: 24th European Photovoltaic Solar Energy Conference, Hamburg Germany, 2009, pp. 21–25.

[25] Stan Benjamin, J Brown, J Olson, J Wilczak, R Banta, G DiMego, F Weng, Improvements in NOAA modeling and data assimilation to improve guidance for the renewable energy industry, in: American Meteorological Society Conf, 2010.

[26] V. Lara-Fanego, J.A. Ruiz-Arias, D. Pozo-Vázquez, F.J. Santos-Alamillos, J. Tovar-Pescador, Evaluation of the WRF model solar irradiance forecasts in Andalusia (southern Spain), Sol. Energy 86 (8) (2012) 2200–2217, Progress in Solar Energy 3.

[27] Richard Perez, Sergey Kivalov, James Schlemmer, Karl Hemker, David Renné, Thomas E. Hoff, Validation of short and medium term operational solar radiation forecasts in the US, Sol. Energy 84 (12) (2010) 2161–2172.

[28] NOAA National Centers for Environmental Prediction (NCEP), NOAA/NCEP Global Forecast System (GFS) Atmospheric Model , 2021, Updated 2023.

[29] National Oceanic and Atmospheric Administration (NOAA), National Digital Forecast Database (NDFD).

[30] Richard Perez, Elke Lorenz, Sophie Pelland, Mark Beauharnois, Glenn Van Knowe, Karl Hemker, Detlev Heinemann, Jan Remund, Stefan C. Müller, Wolfgang Traunmüller, Gerald Steinmauer, David Pozo, Jose A. Ruiz-Arias, Vicente Lara-Fanego, Lourdes Ramirez-Santigosa, Martin Gaston-Romero, Luis M. Pomares, Comparison of numerical weather prediction solar irradiance forecasts in the US, Canada and Europe, Sol. Energy 94 (2013) 305–326.

[31] Akshaya Nikumbh, B. Padmakumari, Sneha Sunil, Cloud fraction retrieval and its variability during daytime from ground-based sky imagery over a tropical station in India, J. Atmos. Sol.-Terr. Phys. 190 (2019) 74–83.

[32] Marion Schroedter-Homscheidt, Hanne Breitkreuz, Carsten Hoyer-Klick, Evangelos Rikos, Stathis Tselepsis, Nowcasting and forecasting of solar irradiance for solar energy electricity grid integration, 2009.
[33] Ricardo Marquez, Carlos F.M. Coimbra, Proposed metric for evaluation of solar forecasting models, J. Sol. Energy Eng. 135 (1) (2013).
[34] C. Renno, F. Petito, A. Gatto, Artificial neural network models for predicting the solar radiation as input of a concentrating photovoltaic system, Energy Convers. Manage. 106 (2015) 999–1012.
[35] Yinghao Chu, Mengying Li, Hugo TC Pedro, Carlos FM Coimbra, Real-time prediction intervals for intra-hour DNI forecasts, Renew. Energy 83 (2015) 234–244.
[36] Ricardo Marquez, Carlos F.M. Coimbra, Intra-hour DNI forecasting based on cloud tracking image analysis, Sol. Energy 91 (2013) 327–336.
[37] Chi Wai Chow, Bryan Urquhart, Matthew Lave, Anthony Dominguez, Jan Kleissl, Janet Shields, Byron Washom, Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed, Sol. Energy 85 (11) (2011) 2881–2893.
[38] E. Crispim, Pedro Ferreira, Antonio Ruano, Prediction of the solar radiation evolution using computational intelligence techniques and cloudiness indices, Int. J. Innovative Comput. Inf. Control 4 (2008) 1121–1133.
[39] Yinghao Chu, Mengying Li, Hugo T.C. Pedro, Carlos F.M. Coimbra, A network of sky imagers for spatial solar irradiance assessment, Renew. Energy 187 (2022) 1009–1019.
[40] Tingting Zhu, Yiren Guo, Zhenye Li, Cong Wang, Solar radiation prediction based on convolution neural network and long short-term memory, Energies 14 (24) (2021) 8498.
[41] Hugo TC Pedro, Carlos FM Coimbra, Mathieu David, Philippe Lauret, Assessment of machine learning techniques for deterministic and probabilistic intra-hour solar forecasts, Renew. Energy 123 (2018) 191–203.
[42] Yinghao Chu, Hugo T.C. Pedro, Carlos F.M. Coimbra, Hybrid intra-hour DNI forecasts with sky image processing enhanced by stochastic learning, Sol. Energy 98 (2013) 592–603.
[43] Enrica Scolari, Fabrizio Sossan, Mathia Haure-Touzé, Mario Paolone, Local estimation of the global horizontal irradiance using an all-sky camera, Sol. Energy 173 (2018) 1225–1235.
[44] Nazle E Herrera-Carrillo, Michel Rivero, Emmanuel Gómez-Ramírez, Rodrigo Loera-Palomo, Solar irradiance estimation based on image analysis, in: 2018 IEEE International Autumn Meeting on Power, Electronics and Computing, ROPEC, IEEE, 2018, pp. 1–6.
[45] Huaiguang Jiang, Yi Gu, Yu Xie, Rui Yang, Yingchen Zhang, Solar irradiance capturing in cloudy sky days–a convolutional neural network based image regression approach, IEEE Access 8 (2020) 22235–22248.
[46] Luis Valentín, Manuel I Peña-Cruz, Daniela Moctezuma, Cesar M Peña-Martínez, Carlos A Pineda-Arellano, Arturo Díaz-Ponce, Towards the development of a low-cost irradiance nowcasting sky imager, Appl. Sci. 9 (6) (2019) 1131.
[47] Jane Oktavia Kamadinata, Tan Lit Ken, Tohru Suwa, Sky image-based solar irradiance prediction methodologies using artificial neural networks, Renew. Energy 134 (2019) 837–845.
[48] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[49] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Trans. Signal Process. 45 (11) (1997) 2673–2681.
[50] Sepp Hochreiter, Jürgen Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, Illia Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, vol. 30, 2017.
[52] Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F Schmidt, Jonathan Weber, Geoffrey I Webb, Lhassane Idoumghar, Pierre-Alain Muller, François Petitjean, Inceptiontime: Finding alexnet for time series classification, Data Min. Knowl. Discov. 34 (6) (2020) 1936–1962.
[53] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, Pierre-Alain Muller, Deep learning for time series classification: a review, Data Min. Knowl. Discov. 33 (4) (2019) 917–963.
[54] Ilya Sutskever, Oriol Vinyals, Quoc V. Le, Sequence to sequence learning with neural networks, 2014,
[55] Alex Graves, Navdeep Jaitly, Abdel-rahman Mohamed, Hybrid speech recognition with deep bidirectional LSTM, in: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, IEEE, 2013, pp. 273–278.
[56] Alex Graves, Sequence transduction with recurrent neural networks, 2012, arXiv preprint arXiv:1211.3711.
[57] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al., Language models are few-shot learners, in: Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 1877–1901.
[58] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, Liang Sun, Transformers in time series: A survey, 2022, arXiv preprint arXiv:2202.07125.
[59] Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016, http://www.deeplearningbook.org.
[60] Aston Zhang, Zachary C. Lipton, Mu Li, Alexander J. Smola, Dive into deep learning, 2021, arXiv preprint arXiv:2106.11342.
[61] Yoshua Bengio, Deep learning of representations for unsupervised and transfer learning, in: Isabelle Guyon, Gideon Dror, Vincent Lemaire, Graham Taylor, Daniel Silver (Eds.), Proceedings of ICML Workshop on Unsupervised and Transfer Learning, in: Proceedings of Machine Learning Research, vol. 27, PMLR, Bellevue, Washington, USA, 2012, pp. 17–36.
[62] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, Distilling the knowledge in a neural network, 2015, arXiv preprint arXiv:1503.02531.
[63] Manajit Sengupta, Yu Xie, Anthony Lopez, Aron Habte, Galen Maclaurin, James Shelby, The National Solar Radiation Data Base (NSRDB), Renew. Sustain. Energy Rev. 89 (2018) 51–60.
[64] Li Fei-Fei, Andrej Karpathy, Justin Johnson, CS231n: Convolutional neural networks for visual recognition, 2023, https://cs231n.github.io.
[65] The pandas development team, pandas-dev/pandas: Pandas, Zenodo, 2020.
[66] Wes McKinney, Data structures for statistical computing in python, in: Stéfan van der Walt, Jarrod Millman (Eds.), Proceedings of the 9th Python in Science Conference, 2010, pp. 56–61.
[67] J.D. Hunter, Matplotlib: A 2D graphics environment, Comput. Sci. Eng. 9 (3) (2007) 90–95.
[68] Diederik P. Kingma, Jimmy Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
[69] Sergey Ioffe, Christian Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, pmlr, 2015, pp. 448–456.
[70] Vinod Nair, Geoffrey E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th International Conference on Machine Learning, ICML-10, 2010, pp. 807–814.
[71] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.
[72] James Bergstra, Yoshua Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (10) (2012) 281–305.

Muhammad Saud Ul Hassan is a visiting researcher at Rice University working on the application of deep learning methods to problems in energy systems and climate science. Concurrently, he is a senior software engineer at Advanced Micro Devices (AMD). Saud has a master's degree in mechanical engineering from Florida State University.

Kashif Liaqat is a Ph.D. student in the Department of Mechanical Engineering at Rice University. He holds a master's degree in mechanical engineering from Florida State University. His areas of research include applying interdisciplinary approaches to energy systems optimization, which encompasses thermal modeling, simulations, artificial intelligence, and software/tool development.

Laura Schaefer, Ph.D. Burton J. and Ann M. McMurtry Chair in Engineering Professor, Mechanical Engineering, Rice University Dr. Schaefer received a B.S. in Mechanical Engineering (1995) and a B.A. in English (1995) from Rice University and her M.S. (1997) and Ph.D. (2000) degrees in Mechanical Engineering from the Georgia Institute of Technology. She was a faculty member in Mechanical Engineering at the University of Pittsburgh from 2000–2015, where she was also a Bicentennial Board of Visitors Faculty Fellow, Deputy Director of the Mascaro Center for Sustainable Innovation, and Associate Director of the Center for Energy. She joined the faculty at Rice in 2015.

Alexander J. Zolan, Ph.D. works at the National Renewable Energy Laboratory as a Researcher in energy systems optimization. His recent work at NREL involves the development of models that seek optimal design, dispatch, and operations of concentrating solar power and hybrid energy systems. As a graduate student at the University of Texas at Austin, Alex collaborated with NREL in the development of an energy resource planning tool that provides optimal design and dispatch decisions for a microgrid built to support the energy needs of a forward-operating base in a remote location. His research experience is in the fields of stochastic optimization and Monte Carlo simulation.