2024 Tenth Indian Control Conference (ICC)
9 - 11 December, 2024
IISER Bhopal, India

Accelerating Solar PV Cell Design Optimization: A Machine Learning-surrogate Hybrid Approach

Ajay Narayanan
Reliance Industries Limited
Bangalore, India
ajay1.narayanan@ril.com

Kuldeep Semwal
Jio Platforms Limited
Bangalore, India
kuldeep.semwal@ril.com

Rahul Nair
Reliance Industries Limited
Cochin, India
rahul.nair@ril.com

Dr. Kedar Kulkarni
Reliance Industries Limited
Bangalore, India
kedar5.kulkarni@ril.com

***Abstract*—Design optimization of solar photovoltaic (PV) cells is challenging due to the complex interplay of numerous parameters and the computationally intensive nature of high-fidelity simulations. This work presents a novel machine learning (ML)-surrogate hybrid optimization framework to address this challenge. Our approach leverages an ML model trained on data generated from a solar PV cell simulator to rapidly predict cell performance based on design parameters. This ML model, a computationally efficient surrogate for the simulator, is then integrated with an optimization algorithm to identify optimal design parameters that maximize the desired PV cell performance metric. The proposed approach is validated through extensive experiments using Quokka 2 as the solar PV cell simulator. Results show that the proposed approach maximizes the chosen cell performance to within 2.1% of that obtained by a rigorous baseline based on Bayesian Optimization, but with a significantly shorter convergence time ($\sim$ 4 times faster).**

***Index Terms*—solar cell design, optimization, machine learning surrogate, solar cell simulation**

I. Introduction

The development of high-efficiency solar PV cells is paramount in accelerating the transition to sustainable energy sources and mitigating the effects of climate change. Therefore, design optimization of solar PV cells has been the focus of wide-spread research [1]. However, the optimization of solar cell designs presents a formidable challenge due to the intricate interplay of numerous parameters, including layer thicknesses, doping concentrations, contact geometries, and material properties. Traditionally, design of new solar PV cells is carried out using a simple trial-and-error approach of manual fabrication, which is expensive and, more importantly, time consuming [2]. High-fidelity physics-based simulations of solar cell performance as a function of its design parameters (θ) expedites design optimization by providing an alternative to manual fabrication. PC1D [3], SCAPS-1D [4], wxAMPS-1D [5], AMPS-1D [6], TCAD [7] and Quokka 2 [8] are some well-known solar cell simulators. However, with the rapid development of high-speed computers and accumulation of data, machine learning (ML) algorithms have the potential to further expedite the design optimization of solar PV cells.

979-8-3503-1416-8/23/$31.00 ©2023 IEEE

Related work: Numerous studies have explored the use of standard optimization algorithms for design optimization of solar PV cells and its components using data generated from numerical and/or physics-based simulations. In [9], Sikder et al. applied the differential evolution (DE) technique to minimize the reflection of a multilayer anti-reflection coatings (ARCs) used in solar PV cells made of TiO2 and nanoporous SiO2. The thicknesses and compositions of individual layers were optimized. In [10], Yin-Jung Chang et.al applied simulated annealing (SA) algorithm to optimize the structure of broadband omnidirectional ARCs for silicon based solar PV cells. In [11], the topology of a light-trapping structure of a solar cell was optimized based on GA method. Thirunavukkarasu et al. [12] demonstrated that there are trade-offs between different electrical parameters of solar PV cells using PC1D [3] simulations for optimization of Al-BSF solar PV cells [13]. Although successful, all of these methods rely on the use of either manual fabrication or rigorous physics-based yet computationally expensive simulations and hence are time-intensive procedures.

Recently, researchers have used ML techniques to predict the cell performance from data obtained either by manual fabrication or physics-based solar cell simulations. For instance, Zhu et al. [14] used artificial neural networks (ANN) [15], trained on simulation data generated using Quokka 2 [8], to predict device power conversion efficiencies. Xiao et al. [16] also used ANN-based learning models to predict power conversion efficiencies of three different silicon solar cell technologies. Nurwaha [17] used instead support vector machines (SVMs) [18] to predict the output power of solar PV cells using data collected experimentally from PV installations. Wagner-Mohnsen et al. [19] compared the performance of linear regression models [20], regression trees [21], SVMs, ensemble of trees [22], and Gaussian process regression (GPR) models [23] to predict the electrical properties of PERC solar PV cells using data generated from simulation that was calibrated using cell measurements.

While ML techniques were shown to be effective to *predict* cell performance parameters, their use in *design optimization* of solar PV cells remains under-explored.

In the context of problems other than solar PV cell design optimization, combining ML surrogates (for computationally expensive simulations) with standard optimization algorithms has been shown to drastically reduce the time required to converge to optimal parameters. For instance, Buratti et al. [24] proposed an ML+GA hybrid approach to optimize different solar PV cell fabrication processes simultaneously. The authors used several ML methods: SVR, RFs [25], adaptive boosting (AB) [26], and NNs to predict the cell efficiency based on process conditions and then a GA was used to identify the combination of input process parameters than can maximize the cell efficiency. In [27], authors have proposed a ML-surrogate hybrid optimization technique for generate a realistic city traffic simulation using the SUMO [28] traffic simulator. The use of ML-surrogate hybrid optimization technique for solar PV cell design optimization is a clear research gap we wish to address in this paper.

In this work, we propose a novel framework that leverages machine learning (ML) to create a surrogate model of a high-fidelity solar PV cell simulator. This surrogate model, trained on data generated by the simulator, can rapidly predict cell performance based on design parameters, effectively bypassing the need for computationally expensive simulations during the optimization process. Our approach offers several key advantages. First, by substituting the computationally demanding simulator with a fast and efficient ML surrogate, we drastically reduce the time and resources required for design optimization. Second, the ML surrogate allows us to explore a vast design space with unprecedented speed, enabling the identification of optimal parameter configurations that may not be easily discovered using traditional methods. Third, our approach is generic in that it can be easily adapted to different solar cell architectures and simulators, making it a versatile tool for accelerating solar PV cell design optimization.

II. Methodology

The proposed ML-surrogate hybrid optimization framework, as shown in Fig 1, is an iterative process designed to identify optimal design parameters for high-performance solar PV cells. Our framework takes as input an initial set of design parameters (θ) and outputs a optimal set of design parameters (θ') which maximises a specific cell performance metric (p). The choice of design parameters (θ) and performance metric (p) will depend on the specific solar cell simulator used. The process begins by generating a matrix (Θ) by randomly sampling values within a $\pm 10\%$ range around the initial set of design parameters (θ). Then, a high-fidelity solar cell simulator is used to generate a dataset (D = Θ,**P**), with each data point consisting of a unique row in Θ and the corresponding simulated solar cell performance metrics (**P**). This dataset (D) serves as the foundation for training a machine learning (ML) model, enabling it to predict solar cell performance metric (p) as a function of design parameters (θ). The trained ML model effectively becomes a surrogate for the computationally demanding solar PV cell simulator.

An optimization algorithm then leverages this ML surrogate to efficiently explore the design space to maximise the ML-predicted cell performance metric (p). The output of the optimization algorithm is a new set of design parameters (θ') that maximises the predicted cell performance metric (p) and its corresponding simulated performance (p'). In the subsequent iterations, the output of the optimization algorithm (θ'), is taken as the starting point for the perturbation process (i.e. $\theta = \theta$') to generate Θ and the process continues. In each iteration, the ML model is retrained exclusively on a new dataset D = Θ,**P**, refining its understanding of the relationship between design parameters (θ) and performance metric (p) in a specific region of the design space. This focused retraining strategy allows for efficient exploration of promising areas while avoiding unnecessary retraining on the entire dataset.

With an improved ML surrogate, the optimization algorithm continues its exploration of the design space and generates the next set of design parameters. This iterative cycle persists until a predefined convergence criterion is satisfied. This criterion could be a maximum number of iterations, a minimum improvement threshold in performance (i.e. difference between p and p'), a minimum change threshold between the design parameters (i.e. the difference between θ and θ') or a combination of all three. The culmination of the ML-surrogate hybrid optimization framework is the identification of design parameters that yield the highest predicted solar cell performance according to the ML surrogate. These parameters can then be rigorously validated through detailed simulations or experimental fabrication to confirm their effectiveness.

III. Results and discussion

A. Study Setup

To evaluate the effectiveness of our proposed framework, we utilized the Quokka 2 solar cell simulator [8]. Quokka 2 is a free simulation software that efficiently models 1D/2D/3D charge carrier transport in silicon solar PV cells with various design options and solver functionalities. Quokka 2 provides a comprehensive set of input parameters for modelling FRC cells, encompassing 172 parameters [29] in total. In this study, we used only the numerical parameters, while keeping the categorical parameters constant. This subset of numerical

TABLE I
COMPARISON OF DIFFERENT OPTIMIZATION ALGORITHMS

Optimization Algorithm	**Baseline PCE**	**Optimal PCE**	**Convergence Time (hours)**
Bayesian Optimization (BO)	17.86%	22.5%	48.0
ML-surrogate Hybrid (GA)		20.52%	11.5
ML-surrogate Hybrid (DE)		20.64%	12.8
ML-surrogate Hybrid (BRKGA)		20.43%	10.2
ML-surrogate Hybrid (ES)		20.38%	13.6
ML-surrogate Hybrid (SRES)		20.45%	12.1

TABLE II
COMPARISON OF DIFFERENT ML MODELS FOR PCE PREDICTION

Parameter	**Model Name**	**CV RMSE**	**Test RMSE**	**MAPE**
PCE	Linear Regression	0.053785	0.057328	2.93%
	Random Forest	0.089761	0.088313	4.44%
	Gradient Boosting	0.050081	0.051023	2.61%

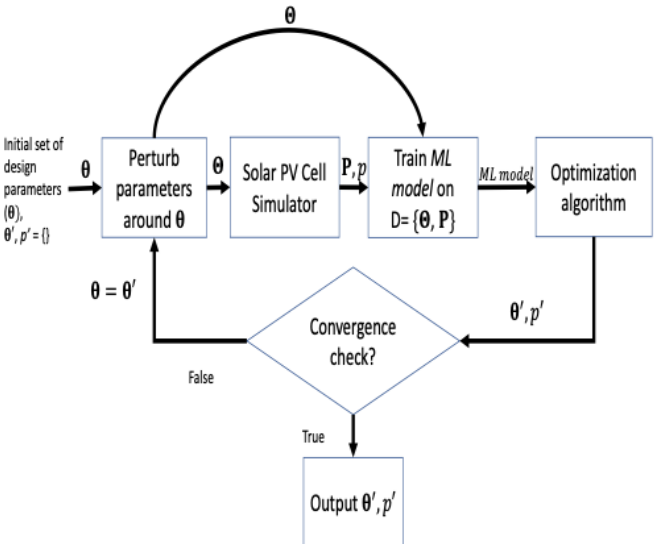


Fig. 1. Proposed methodology

parameters, which includes layer thicknesses, doping concentrations, and contact dimensions, served as the design parameters (θ) for our ML-surrogate hybrid optimization framework, and the cell power conversion efficiency (PCE) [30] was chosen as the cell performance metric (p) to be maximized. In each iteration, 1000 permutations of the design parameters (Θ) were generated by perturbing the optimized parameters from the previous iteration within a range of $\pm 10\%$. These perturbed settings were simulated using Quokka 2, and the resulting cell performance data was used to retrain the ML model, as discussed in section 2.

In this study, we train and evaluate three different machine learning (ML) algorithms for constructing the surrogate model: linear regression [20], random forest [25], and gradient boosting [31]. The dataset (D) generated using the Quokka 2 simulation was split into training (80%) and test sets (20%). The numerical features were scaled using the StandardScalar [32] and all models were trained using 5-fold cross-validation [33]. The performance of each ML model was assessed based on its ability to predict the power conversion efficiency (PCE) [30] as the cell performance metric (p).

We explored five different algorithms for optimization: genetic algorithm (GA) [34], differential evolution (DE) [35], biased random key genetic algorithm (BRKGA) [36], evolutionary strategy (ES) [37], and stochastic ranking evolutionary strategy (SRES) [38]. The performance of each algorithm, when coupled with the ML-surrogate, was compared against a rigorous, baseline Bayesian optimization (BO) approach [39] that directly used the Quokka 2 simulator as a black box, as opposed to an ML-surrogate. The BO approach serves as a benchmark for evaluating the accuracy as well as the computational efficiency of our ML-surrogate hybrid framework. The algorithms are assessed based on their ability to maximize PCE, while also considering the computational time required for convergence. The highest simulated PCE in the dataset (D) generated in the 1st iteration is taken as the **baseline PCE** to benchmark the performance of each optimization algorithm.

B. Performance of different ML models

Table II summarizes the performances of the various ML models that were trained and tested for PCE prediction. Training time for all models was a few seconds. Models were evaluated using the root mean square error (RMSE) and the mean absolute percentage error (MAPE) as metrics. In general, all models performed well ($\leq$5% MAPE). Among the 3 models tested, the GradientBoosting model showed the best performance with the lowest test RMSE (0.051) and MAPE (2.61%). These findings are further validated in Fig 2, which depicts the scatter plots of predicted (by the ML model) versus actual values of PCE. The scatter plots reveal strong correlations for Linear Regression and Gradient Boosting, indicating their effectiveness in capturing the underlying relationships. Random Forest demonstrates reasonable accuracy but with a slightly wider spread of predictions compared to the other two models.

C. Performance of different optimization techniques

Table I summarizes the results of the different optimization algorithms considered. As evident from Table I, the Bayesian Optimization (BO) approach, using the Quokka 2 simulator directly, yields the highest optimal PCE of 22.5% (an increase of 24.93% over the baseline PCE) but at the cost of a substantial computational time of 48 hours. In contrast, the ML-surrogate hybrid with Differential Evolution (DE) achieves a optimal PCE of 20.64% (2.1% lower than BO and 15.57% higher than the baseline PCE) while converging $\sim$3.75 times faster than BO. In general, all of the ML-surrogate Hybrid were able to show significant improvement ($\geq$13% increase) from the baseline PCE. These results highlight the potential of our proposed ML-surrogate hybrid framework for accelerating solar cell design optimization.

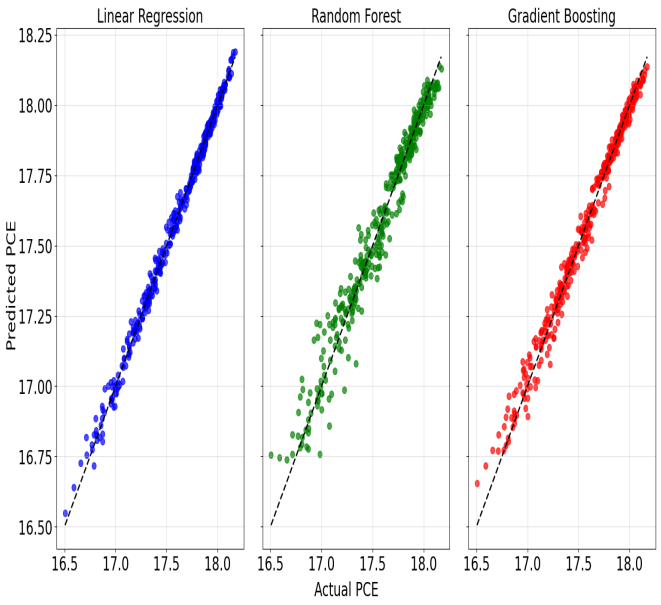


Fig. 2. Scatter plots of predicted vs. actual PCE

D. Performance of the ML-surrogate hybrid optimization framework

To further analyse our ML-surrogate hybrid optimization approach, we visualized the optimization trajectory followed by DE (best performing optimization algorithm as discussed in section 3.3) while using the GradientBoosting (best performing ML model as discussed in section 3.2) model as a surrogate, as shown in Fig 3. The green dots show the highest simulated PCE (using Quokka 2) in the training dataset (D) at the beginning of each iteration, referred to as the baseline PCE for each iteration. The black line depicts the optimization trajectory followed by the ML-surrogate model over 500 generations in each iteration. The red stars indicate the simulated PCE (using Quokka 2) of optimal parameters (p') received as output from the DE at each iteration. In this experiment, the convergence criteria was defined in terms of the maximum number of iterations, which was taken to be 14. The figure demonstrates a consistent improvement in PCE across the 14 iterations, starting from an initial baseline PCE of 17.86% and reaching a final efficiency of 20.64%, representing a **15% overall improvement**.Another interesting observation is that the simulated PCE of the optimal parameters (p') produced by the DE (red stars) are consistently higher than what was predicted by the ML surrogate.

To further investigate the ML-surrogate hybrid optimization, we plot the difference between p' (simulated PCE of the optimal parameters obtained in the current iteration) and p (simulated PCE of the optimal parameters in the previous iteration) across the 14 iterations , as shown in Fig 4. As

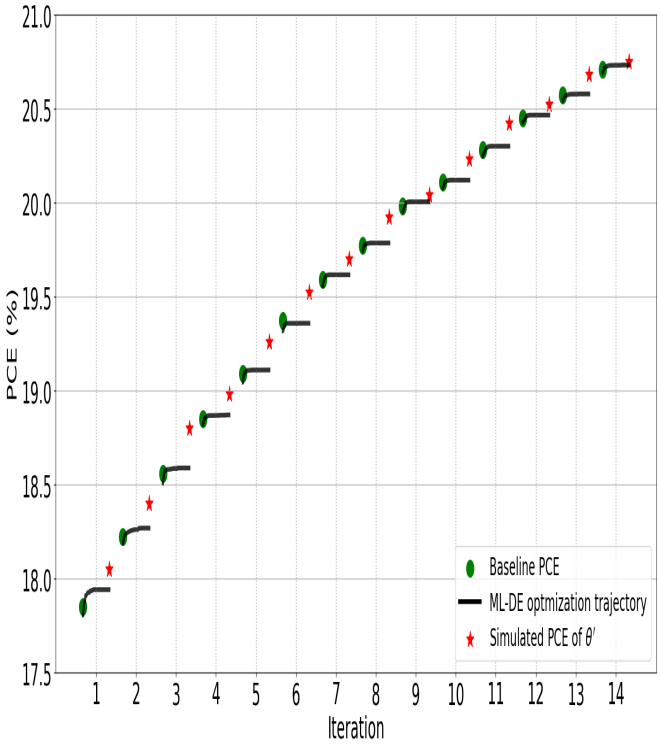


Fig. 3. Evolution of PCE Over Successive Optimization Iterations

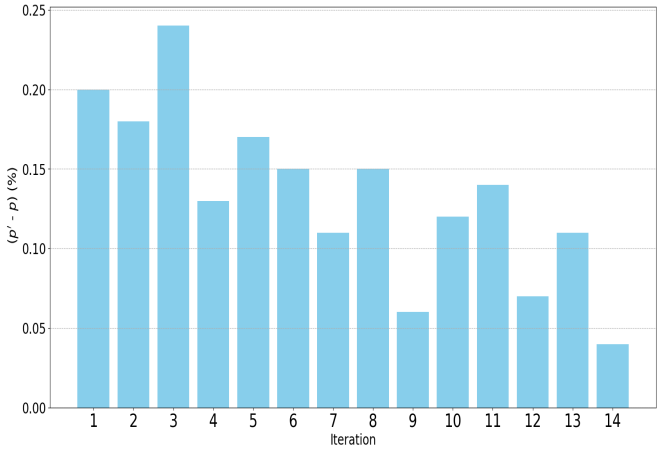


Fig. 4. *(p' - p)* across iterations

in evident from the figure, the quantity *(p' - p)* exhibits fluctuation throughout the iterations, reaching a peak at the 3rd iteration and generally decreasing with some variations as iterations progress. Notably, higher gains in PCE were observed in the early iterations, with a gradual tapering off in improvement as the optimization process progressed.

Fig 5 illustrates the Root Mean Square Error (RMSE) loss of the trained Gradient Boosting (GB) model over 14 iterations. The RMSE loss generally decreases as the number

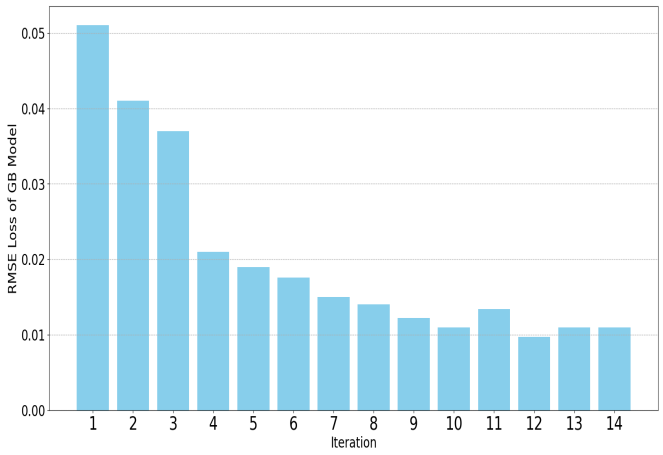


Fig. 5. RMSE loss of gradient boosting (GB) model across iterations

of iterations increases. This suggests that the ML model's predictive ability gradually improves as the optimization process progresses. This is an artefact of the focused retraining strategy described in section 2. In each iteration, since the ML model is retrained exclusively on a newly generated dataset, it is able to refine its understanding of the relationship between design parameters (θ) and performance metric (p) in a specific region of the design space, as evidenced by Fig 5.

IV. Conclusion

In this work, we demonstrated the potential of a novel ML-surrogate hybrid optimization framework for accelerating the design of high-performance solar PV cells. The proposed framework was thoroughly evaluated in extensive experiments using the Quokka 2 solar cell simulator. Our results showed that the ML-surrogate hybrid approach, could achieve a PCE comparable ($\sim$ 2.1% lower) to that of Bayesian Optimization but with a significantly shorter convergence time ($\sim$ 4 times faster). The effectiveness of our framework in rapidly identifying promising design parameters highlights its potential for accelerating the development of high-performance solar PV cells and advancing photovoltaic technology. Future work will focus on extending this approach to other solar cell architectures, incorporating more sophisticated ML models, and exploring multi-objective optimization to address the potential trade-off between different cell performance metrics.

References

[1] W. Issaadi, S. Issaadi, and A. Khireddine, "Comparative study of photovoltaic system optimization techniques: Contribution to the improvement and development of new approaches," *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 2112–2127, 2018.

[2] S. Bhatti, H. U. Manzoor, B. Michel, R. S. Bonilla, R. Abrams, A. Zoha, S. Hussain, and R. Ghannam, "Revolutionizing low-cost solar cells with machine learning: a systematic review of optimization techniques," *Advanced Energy and Sustainability Research*, vol. 4, no. 10, p. 2300004, 2023.

[3] D. A. Clugston and P. A. Basore, "Pc1d version 5: 32-bit solar cell modeling on personal computers," in *Conference Record of the Twenty Sixth IEEE Photovoltaic Specialists Conference-1997*. IEEE, 1997, pp. 207–210.

[4] M. Mostefaoui, H. Mazari, S. Khelifi, A. Bouraiou, and R. Dabou, "Simulation of high efficiency cigs solar cells with scaps-1d software," *Energy Procedia*, vol. 74, pp. 736–744, 2015.

[5] Y. Liu, Y. Sun, and A. Rockett, "A new simulation software of solar cells—wxamps," *Solar Energy Materials and Solar Cells*, vol. 98, pp. 124–128, 2012.

[6] H. Zhu, A. K. Kalkan, J. Hou, and S. J. Fonash, "Applications of amps-1d for solar cell simulation," in *AIP Conference Proceedings*, vol. 462, no. 1. American Institute of Physics, 1999, pp. 309–314.

[7] Y.-C. Wu, Y.-R. Jhan, Y.-C. Wu, and Y.-R. Jhan, "Introduction of synopsys sentaurus tcad simulation," *3D TCAD Simulation for CMOS Nanoeletronic Devices*, pp. 1–17, 2018.

[8] A. Fell, "A free and fast three-dimensional/two-dimensional solar cell simulator featuring conductive boundary and quasi-neutrality approximations," *IEEE Transactions on Electron Devices*, vol. 60, no. 2, pp. 733–738, 2012.

[9] U. Sikder and M. A. Zaman, "Optimization of multilayer antireflection coating for photovoltaic applications," *Optics & Laser Technology*, vol. 79, pp. 88–94, 2016.

[10] Y.-J. Chang and Y.-T. Chen, "Broadband omnidirectional antireflection coatings for metal-backed solar cells optimized using simulated annealing algorithm incorporated with solar spectrum," *Optics Express*, vol. 19, no. 104, pp. A875–A887, 2011.

[11] C. Wang, S. Yu, W. Chen, and C. Sun, "Highly efficient light-trapping structure design inspired by natural evolution," *Scientific reports*, vol. 3, no. 1, p. 1025, 2013.

[12] G. S. Thirunavukkarasu, M. Seyedmahmoudian, J. Chandran, A. Stojcevski, M. Subramanian, R. Marnadu, S. Alfaify, and M. Shkir, "Optimization of mono-crystalline silicon solar cell devices using pc1d simulation," *Energies*, vol. 14, no. 16, p. 4986, 2021.

[13] S. Sepeai, S. Cheow, M. Sulaiman, K. Sopian, and S. H. Zaidi, "Fabrication and characterization of al-bsf bifacial solar cell," in *2013 IEEE 39th Photovoltaic Specialists Conference (PVSC)*. IEEE, 2013, pp. 2664–2668.

[14] H. Zhu, W. Yan, Y. Liu, D. Hu, Y. Tu, Z. Huang, and X. Tan, "Design investigation on 100 μm-thickness thin silicon perc solar cells with assistance of machine learning," *Materials Science in Semiconductor Processing*, vol. 137, p. 106198, 2022.

[15] S. Haykin, *Neural networks and learning machines, 3/E*. Pearson Education India, 2009.

[16] W. Xiao, G. Nazario, H. Wu, H. Zhang, and F. Cheng, "A neural network based computational model to predict the output power of different types of photovoltaic cells," *PloS one*, vol. 12, no. 9, p. e0184561, 2017.

[17] D. Nurwaha, "Comparison of kernel functions of support vector machines: A case study for the solar cell output power prediction," *International Journal of Energy Applications and Technologies*, vol. 7, no. 1, pp. 1–6, 2020.

[18] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, pp. 199–222, 2004.

[19] H. Wagner-Mohnsen and P. P. Altermatt, "A combined numerical modeling and machine learning approach for optimization of mass-produced industrial solar cells," *IEEE Journal of Photovoltaics*, vol. 10, no. 5, pp. 1441–1447, 2020.

[20] S. Weisberg, *Applied linear regression*. John Wiley & Sons, 2005, vol. 528.

[21] W.-Y. Loh, "Classification and regression trees," *Wiley interdisciplinary reviews: data mining and knowledge discovery*, vol. 1, no. 1, pp. 14–23, 2011.

[22] Y. Ren, L. Zhang, and P. N. Suganthan, "Ensemble classification and regression-recent developments, applications and future directions," *IEEE Computational intelligence magazine*, vol. 11, no. 1, pp. 41–53, 2016.

[23] E. L. Snelson, *Flexible and efficient Gaussian process models for machine learning*. University of London, University College London (United Kingdom), 2008.

[24] Y. Buratti, C. Eijkens, and Z. Hameiri, "Optimization of solar cell production lines using neural networks and genetic algorithms," *ACS Applied Energy Materials*, vol. 3, no. 11, pp. 10 317–10 322, 2020.

[25] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[26] H. Schwenk and Y. Bengio, "Training methods for adaptive boosting of neural networks," *Advances in neural information processing systems*, vol. 10, 1997.

[27] A. Narayanan, R. Nair, and K. Kulkarni, "A city-agnostic framework to determine optimal location and sizing of electric vehicle charge stations," in *Proceedings of the 7th Joint International Conference on Data Science & Management of Data (11th ACM IKDD CODS and 29th COMAD)*, ser. CODS-COMAD '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 523–527. [Online]. Available: https://doi.org/10.1145/3632410.3632476

[28] D. Krajzewicz, "Traffic simulation with sumo–simulation of urban mobility," *Fundamentals of traffic simulation*, pp. 269–293, 2010.

[29] "Quokka 2 Manual — www2.pvlighthouse.com.au," [Accessed 14-06-2024].

[30] G. P. Smestad, F. C. Krebs, C. M. Lampert, C. G. Granqvist, K. Chopra, X. Mathew, and H. Takakura, "Reporting solar cell efficiencies in solar energy materials and solar cells," pp. 371–373, 2008.

[31] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers in neurorobotics*, vol. 7, p. 21, 2013.

[32] "StandardScaler — scikit-learn.org," [Accessed 14-06-2024].

[33] "3.1. Cross-validation: evaluating estimator performance — scikit-learn.org," [Accessed 14-06-2024].

[34] A. Lambora, K. Gupta, and K. Chopra, "Genetic algorithm-a literature review," in *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon)*. IEEE, 2019, pp. 380–384.

[35] V. Feoktistov, *Differential evolution*. Springer, 2006.

[36] J. F. Gonçalves and M. G. Resende, "Biased random-key genetic algorithms for combinatorial optimization," *Journal of Heuristics*, vol. 17, no. 5, pp. 487–525, 2011.

[37] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies–a comprehensive introduction," *Natural computing*, vol. 1, pp. 3–52, 2002.

[38] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on evolutionary computation*, vol. 4, no. 3, pp. 284–294, 2000.

[39] P. I. Frazier, "Bayesian optimization," in *Recent advances in optimization and modeling of contemporary problems*. Informs, 2018, pp. 255–278.