Recent Advances in Silicon Solar Cell Research Using Data Science-Based Learning

Rahul Jaiswal, Manel Martínez-Ramón, and Tito Busani

***Abstract*—The application of machine learning techniques in silicon photovoltaics research and production has been gaining traction. Learning from the existing data has given the potential to research labs and industries of discovering optimized processing parameters, device architectures, and fabrication recipes. It has also been utilized for defect detection and quality inspection. The increasing computational capacities of modern computers and abstraction of machine learning algorithms, along with the increasing community support for open-source software libraries has increased the accessibility of learning-based algorithms that were perceived as complex to be implemented for interdisciplinary research and development just a few years back. In this article, we present a review of the efforts in the literature that have utilized machine learning techniques for commercial silicon solar cell devices in recent times. The emphasis is to categorize and investigate specific learning techniques that are best suited for one particular device or fabrication process parameter optimization. We also provide insight into possible expansions of current research methodologies that can further improve the prediction accuracy while minimizing the computational costs and extract other useful information from a machine learning model, such as prediction uncertainty, scalability, and generalization of a particular model.**

***Index Terms*—Classification, computer vision, machine learning (Ml), regression, solar energy.**

I. Introduction

Silicon solar cells continue to have the largest share in commercial solar energy market, therefore, efforts in research to improve individual cell power conversion efficiencies are still being pursued. These efforts are directed toward either improving the performance of the cell (for example, its electrical performance like power conversion efficiencies or optical performance like reflection losses), or improving the fabrication processes and recipes or to reduce defects and losses in the device. Conventional strategies for designing and optimizing a photovoltaic (PV) device can be inefficient in terms of resources and manpower as they involve doing the design of experiments and even disruption of production lines in case of industrial research, another issue with conventional optimization strategies is that the transition time-period between a lab prototype to large scale fabrication can be very long [1], [2]. Data driven decision making [3] provides an obvious alternative approach for achieving this target with a better efficiency as it can supplement the conventional knowledge about the device with a forecast about the nature of the device. This can save unnecessary or redundant experiments (i.e., either fabricating physical devices or performing computational expensive simulations). Zhu et al. [4] describe one such work where the performance of PERC cells with a thin substrate is investigated using multilayer perceptron predictions. Fig. 1 illustrates this trend.

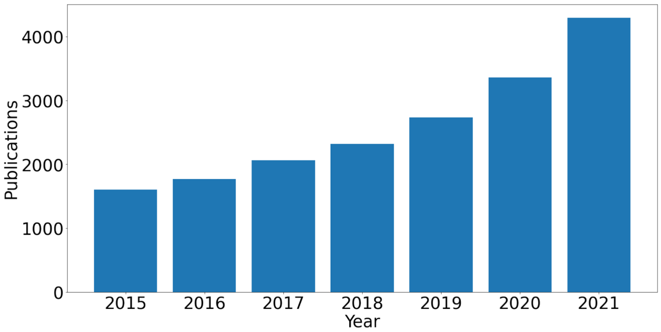

Fig. 1. Number of publications indexed by the keywords "silicon," "solar," "cell," and "Ml."

Graphics processing units (GPU) use single instruction, multiple data architecture, which allows parallel computing or task concurrency for the computations required in machine learning (Ml) algorithm training and inference. This makes them more suitable for Ml compared to CPUs. This is also reflected in the literature by the increasing complexity of the algorithms being implemented and the data size used for training in the recent past.

The advances in hardware implementations suitable for deep learning (DL) [5] have come along with advances in the software written for machine ML. Particularly, Python is by large the most used programming language for DL, not only due to its object-oriented design and flexible nature, but also because it can run on any operating system and it is open source.

Manuscript received 6 July 2022; revised 23 August 2022 and 11 October 2022; accepted 4 November 2022. Date of publication 12 December 2022; date of current version 12 January 2023. The work of Manel Martínez-Ramón was supported in part by the National Science Foundation EPSCoR under Grant OIA-1757207 and in part by the King Felipe VI endowed Chair. *(Corresponding author: Tito Busani.)*

Rahul Jaiswal and Tito Busani are with the Center for High Technology Materials, University of New Mexico, Albuquerque, NM 87131-0001 USA, and also with the Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87131-0001 USA (e-mail: rahul17455@gmail.com; busanit@unm.edu).

Manel Martínez-Ramón is with the Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87131-0001 USA (e-mail: manel@unm.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JPHOTOV.2022.3221003.

Digital Object Identifier 10.1109/JPHOTOV.2022.3221003

Many open-source libraries have been developed to program DL structures that can be used in parallel programming hardware, particularly GPUs. Theano [6] is probably the first open-source library for Ml that integrates other standard scientific libraries as NuMpy and SciPY. Maybe the paradigm of this is TensorFlow [7], which also include differentiable programming tools or, in other words, the capability to automatically compute the back propagation of complex DL structures with autogradient techniques, which makes implementation of these DL structures much easier. Package Keras [8] works on top of TensorFlow to add more abstraction and compactness to the programming. Pytorch [9] is an alternative to TensorFlow that trades part of the TensorFlow flexibility by an easier implementation of DL structures, which makes it suitable for DL training. It also includes autograd and it is known to be fast.

Implementation of other ML techniques, such as ensemble learning are also becoming accessible through open-source implementations, such as lightGBM [10] and XGBoost [11]. Scikit-learn [12] library in Python, on the other hand, has provided a very abstract way to fast prototype multiple learning models except DL. These tools have provided increased flexibility to researchers for exploring data-based learning.

Mass produced silicon-based PVs technologies, such as aluminium back surface field (Al-BSF) [13], passivated emitter and rear contact (PERC) [1], and Heterojunction with thin intrinsic layer [14] solar cells are comparatively mature technologies and any information gathered using learning-based techniques to improve an existing device can be verified by incorporating the predicted information in the device design and fabrication processes. Compared to novel PVs technologies, this leads to an easy validation of the prediction model on a physical device and model prediction uncertainty can be documented when the device is fabricated on a large scale.

This review will be covering the different learning algorithms that have been used in the literature, their prediction accuracy and computational complexity. Application of similar learning algorithms in multiple articles will be compared to create an information library that can be used for a future exploration.

The application of learning algorithms in PVs device optimization is often focused on optimizing a single device or process parameter. This is achieved by identifying the critical loss mechanisms, and then reducing these losses. Although this is a complex task as many loss mechanisms within the device are interrelated [15] and trying to reduce one loss may increase another. Therefore, a major emphasis of any published work should be to address these tradeoffs. One such example is in Wagner-Mohnsen and Altermatt [16], where interdependent parameter bias and tradeoffs are discovered in a PERC solar cell.

For solving this issue of finding the correlations between different parameters that are associated with a solar cell device, the prediction and classification mechanism of conventional Ml implementation strategies have to be specifically tailored for a given solar cell architecture. The uncertainty of a given prediction, its scalability and the potential of generalization should also be discussed in scholarly work. Burrati et al. [17] discuss these points in their implementation of ensemble techniques to extract bulk defects. Throughout this review, we will be highlighting such efforts already implemented in the literature, and also discussing their necessity in a published work where they have not been discussed.

This review is divided into three main sections where application of Ml algorithms for device optimization, process optimization, and defect analysis will be discussed in detail.

II. Application of ML Techniques Implemented for Optimization of Device Performance

Learning methods can be used to optimize either a specific process parameter involved in device fabrication or they can be used to optimize the design of a solar cell (which can consist of multiple parameters and design features, such as its architecture, material composition, or specific material property). In this section, literary works that have utilized learning algorithms for device optimization are discussed and are summarized in Table I.

A. Prediction and Optimization of Solar Cell Electrical Parameter

As per the International Roadmap for PVs 2021 report [18], PERC solar cells had a market share of 80% in 2020. Al-BSF solar cells had a market share of 18% and silicon heterojunction (SHJ) solar cells installation capacity was around 1%–2%. It is projected that SHJ solar cells market share will increase in the future. Efforts in academia and industry to optimize the electrical performance metrics of these industrial solar cell devices are available using conventional strategies, such as Thirunavukkarasu et al. [19] work to use PC1D [20] simulations for optimization of Al-BSF solar cells. This work demonstrates that there are tradeoffs between different electrical parameters of the device, so any design and material change (for example, wafer thickness and doping, respectively) has to find an optimal value to maximize the final device performance by power conversion efficiency (PCE, also denoted by η), while allowing the device to be scalable for production (for example, wafer thickness less than 150 μm is result in better device efficiency but handling such wafer is difficult in this case). One can use simulation-based learning only to achieve such optimizations, but each of these learnings will be relevant to only a singular device architecture and design and it can be slow. Data-based learning on the other hand can potentially generalize each research work for other use cases while requiring less time to execute. Such learning techniques to optimize device electrical performance are available in the literature and are discussed as follows.

Zhu et al. [4] investigated the performance of PERC cell with thin substrates. Three test device designs with thin substrates (100 μm) were simulated using the Quokka simulation tool [21] and the simulation data was used to train artificial neural networks (ANN) [22] to predict device power conversion efficiencies. The trained Ml model was used to investigate the ranking of these thin films according to the importance of the efficiency gain. Results show that the thickness of thin layers at the front and rear texturing are correlated with the performance of the PERC devices, specifically the silicon dioxide film, which

TABLE I
SUMMARY OF SECTION II

Prediction domain	**Article**	**Cell technology**	**Target**	**Model used**	**Dataset availability**	**Features**	**Critical parameter or outcome**
Electrical performance	[4]	PERC cells	η	ANN	Included in supplementary data	Emitter, bulk, and BSF parameters	Front thin layer thickness
	[23]	Mono, multi crystalline, and amorphous cells	η	ANN	Included in supporting information	External factors (temperature, light intensity)	No. of layers in ANN
	[25]	Cell technology not specified	P_{MPP}	SVM	Not provided	Ambient weather condition	SVM implementation strategy and kernels used
	[16]	PERC	η, V_{OC}, P_{MPP}, V_{MPP}	SVM, GPR, Ensemble, linear regression	Simulation methodology provided	Wafer resistivity, dopant profiles, lifetime, finger dimension, BSF depth, passivation quality	Interdependent parameter bias and tradeoffs
	[26]	Monocrystalline (mc)	V_{OC}, j_0	U-Net	Raw data not provided	Image (pixel level) data	j_0 beneath metal finger prediction
Optical performance	[27]	Thin film a-Si cell	QE	ANN	Available on request from authors	Light wavelength andcell cell geometry	Optimal ANN structure
	[28]	Cell technology not provided	Optical losses	ANN	PC1D simulations	ARC thickness, device thickness, rear reflection	ARC thickness
Diode model parameters	[29]	Cell technology not pro- vided	I_L, I_S, R_S, R_{SH}, N	Multitarget SVM, RF, and regression trees	Simulated (one diode model)	Current–voltage profile	I_L prediction accuracy is highest
	[30]	Polycrystalline solar cells	I_L, I_S, , R_S, R_{SH}, N	Bayesian regularizer and ANN	Simulated (one diode model)	Current–voltage profile	Learning is transferable
	[31]	Cell technology not provided	I_L, I_{S1}, I_{S2}, R_S, R_{SH}	GA and PSO	Simulated (two diode model)	Current–voltage profile	PSO convergence was faster
	[32]	Flexible amorphous silicon solar cell	$I_L, \mu_\tau, V_{bi}, d_i$, R_S, R_{SH}, N, I_S	FA	Simulated (one diode model)	Current–voltage profile	FA performed best amongst evolutionary algorithm

provides a good passivation for the device. The authors of [4] had justified the use of ANN by elaborating its generalization capabilities and its superiority in handling highly correlated data. Nevertheless, no empirical proof of performance comparison between other learning algorithms was provided. The network architecture of ANN and was also not discussed in work [4], although input dataset and features were provided in detail in supplementary material. As the dataset size used for training was small (139 training data points and 17 test points), model generalization also cannot be established quantitatively. This research can be further investigated by comparing the prediction accuracy for other learning algorithms, by evaluating if the model has any high variance or bias and by increasing the model generalization.

Xiao et al. [23] had also used ANN-based learning models to predict power conversion efficiencies of the following three silicon solar cell technologies: mono-crystalline, multicrystalline, and amorphous crystalline [24]. Training data was created from experimental solar cell samples by characterizing them under different test conditions of incident light intensity and ambient temperature. Out of the 72 data points, 62 points were used to train the model, and 10 were used to test the model. The prediction results were in agreement with conventional physics-based studies of solar cells. The findings also included a suggestion for recommended number of hidden layers in the ANN architecture for the three different solar cell technologies. The importance of the number of hidden layers in an ANN network for quality of predictions was also discussed in detail by the authors.

Nurwaha [25] used instead support vector machines (SVMs) [33] to predict the output power of solar cells. The author tested four kernels [34]: radial basis kernel, sigmoid kernel function, linear function, and polynomial kernel function. Data were collected experimentally from PV installations. Two support vector regression (SVR) implementations had been tested: epsilon-SVR [35] and nu-SVR [36]. Results show that the nu-SVR with radial basis function kernel and epsilon-SVR with polynomial kernel function had the best accuracies among the tested kernel functions and SVR implementations. The author has correctly pointed out that there are very few published works in the literature that leverage SVMs for PVs research, this work can be extended by a detailed discussion of the model implementation and the feature values and their distributions.

Wagner-Mohnsen et al. [16] have compared the performance of linear regression models [37], regression trees [38], SVMs, ensemble of trees [39], and Gaussian process regression (GPR) models [40] to predict the electrical properties of PERC solar cells. The training data for these Ml models was generated from simulation that was calibrated using cell measurements. This work shows that an ML model that has high accuracy to predict solar cell performance can be used to generate a large volume dataset to infer device properties and discover interdependent relations between different parameters. The same analysis can be very expensive in terms of computational time and resources if done through either experimental approaches or a simulation-based approach. Compared to similar works in the literature, the authors have provided a detailed insight into the applicability of a learning model to understand interdependent parameter bias that exists in a PERC device and the correlation between a device parameter to a performance metric of the device. The methodology of this article can be extended to other solar cell architectures as the learning is general and not unique to the PERC cell.

Learning from images using DL [41] and computer vision [42] can be used in PVs research to investigate the spatial features of a device. Kovvali et al. [26] had investigated the use of DL in their work to infer the material quality of the solar cell device. They had trained the Ml model to predict locally distributed dark saturation current density (j_0). The input of the prediction model were photoluminescence (PL) images of as-cut wafers. Using analytical expressions (1-diode model), an implied open-circuit voltage (V_{OC}) was also calculated from the spatial j_0 distribution. The authors had to modify the U-Net convolution neural network (CNN) [43] for doing regression analysis. This work provides a fast way to analyze the j_0 values at metal contacts from the PL images of as-cut wafers, giving foresight into the final device performance and with a reported prediction time of one microsecond, this methodology can be potentially used for in-line inspection of wafers. This article fundamentally uses PL images as a data source to predict saturation current density values under metal contacts. If this valued had to be calculated using experimental measurements, it would require fabrication of precursor samples. The total saturation current density in the solar cell device is the sum of saturation current density from the emitter, the bulk, and the rear. In order to measure the metal contact saturation current density, two precursor samples are required, one test sample which will have contact just on single side and as well as a lifetime sample, which does not have contacts at either side. From these two samples, the saturation current density values at the emitter can be measured separately. By plotting the fraction of the front surface metal contact against the sum of the two emitter saturation current density values, a fit for j_0 can be obtained. This ordeal of measuring the saturation current density can take up few days as it involves the fabrication of multiple test samples (with different fraction of metal contacts) and disruptions of pilot line. This is in stark contrast to the work presented in [26] that uses a wafer luminescence image (that is part of the process flow) and predict values within seconds.

B. Prediction and Optimization of Solar Cell Optical Performance

The optical performance of a solar cell is a precursor to calculate the final power conversion efficiency of the device, it can be optimized experimentally (for example, one approach can be to fabricate cells with different antireflective layer properties followed by doing characterization to find a fit for optimal parameters) or by simulations (prototyping different design changes using analytical system of formulas or numerical simulation methods, such as ray tracing or transfer matrix). Kang et al. [44] presented an in-depth investigation of silicon nitride antireflection coatings (ARC) using simulations. The optical performance was evaluated using ray tracing and then the optical profile was fed into PC1D for device simulation. Planar and texture solar cells were investigated under different encapsulations, and the study reported optimal refractive index value for the ARC to maximize the PCE. Although this work has provided an extensive simulation dataset to find a fit for ARC parameters to achieve an optimal optical performance, transferring this knowledge for another solar cell architecture and encapsulation materials will require performing the simulations again, which can be time and computational power intensive (specifically ray tracing algorithms), ML methods, on the other hand, can utilize these kind of work in the literature as foundation and can provide an alternative approach for device optimization. One use-case where ML methods can be particularly useful is when the input data (design parameters) is changed within a range where the model was trained. Interpolation for the target (predicted) values can be achieved with good accuracy. Several works in the literature using data-based learning to the optimize optical performance of solar cells are presented as follows.

Kaya et al. [27] demonstrated a prototyping technique for finding the optical performance of thin film solar cells using neural networks. The training dataset was created by using the finite-difference time-domain (FDTD) simulation methods [45]. Results showed that the neural network (NN)-based methodology to predict the optical absorption as a function of incident light wavelength, and the cell geometry was accurate and much faster compared to FDTD simulations. The authors have explained why the conventional simulation approach to calculate optical performance metrics will be slower compared to an ML-based equivalent model that predicts this information, as the former cannot be solved using closed form solutions and require iterative simulations. The authors also provided insight into why a learning-based approach can predict an optical parameter at a given frequency. The decision to choose neural networks over other algorithms was justified by the virtue of its ability to learn functions that are nonlinear, although no empirical comparison was provided. The authors provided information about how the training data was generated and the reason for choosing the quasi-Newton optimization algorithm [46]. They also provided information about the six features of the NN input and its three-layer architecture. This work explores the fundamental reasons behind the ability of an ML-based surrogate model to approximate the results of a device physics-based simulation model. This research is a knowledge base that can be extended further by investigating other learning-based models as, for example, a Bayesian optimization [47] strategy to infer parameters that can maximize the quantum efficiency (QE) of given device architecture.

Shivhare et al. presented a work [28] to predict the thickness of the ARC using ANN's in a cell to achieve reduced optical losses. The authors used PC1D, a 1-D numerical simulation tool to create the training database. Voltage and current points are used as input to the model to predict the required thickness to achieve this performance. It is indicated that the ANN was trained for 1000 iterations, it was reported that to achieve more power output from a solar cell the thickness of the ARC coating should be increased. Future research can explore other physical parameters that can affect the optical performance of the device (such as rear reflection and device thickness). Efforts to find correlation between just the voltage and current and ARC thickness can also be established, this will provide further generalization of presented work for other solar cell designs. A detailed discussion of the exact architecture of the neural network can also be included in such future works.

C. Prediction of Solar Cell Diode Model Parameters

Single and double diode models provide an abstract way to prototype solar cell performance quickly. These models can also be scaled to evaluate the module level performance. They require fitting parameters in the diode model to match the electrical characteristics of the cell. Fitting (linear or nonlinear) based methodologies are available in the literature. In the one presented by Caracciolo et al. [48], the goal was to fit one model parameter at a time using least square methods. Another approach to fit diode model parameters can be using ML models that can perform multiparameter regression as multitask SVMs [49] or multitask Gaussian processes (GPs) [50], [51], [52], [53], few of these efforts are presented as follows.

Shah [29] used multiparameter regression methods to infer multiple parameters of a single-diode solar cell model. Multitarget SVR [54], multitarget regression trees [55], and multitarget random forest (RF) regression [56] were the three regression algorithms applied in this work. A total of 10 000 single diode simulations were used to create current–voltage profiles of 25 data points each using a set of 5 parameters in the simulation—reverse saturation current (I_S), light generated current (I_L), ideality factor (N), series resistance (R_S), shunt resistance (R_{SH}). The current–voltage profile was used as the input to the Ml model to predict the five parameters of the diode model. The author compared the three regression strategies-based on the mean absolute percentage error [57] by predicting diode model parameters from experimentally measured current–voltage profile. The authors also provided suggestions to improve further by using a larger training dataset and the possibility to expand this work for a two diode model is also discussed.

Cortés et al. [30] also investigated a method to estimate parameters of a single-diode model for a PV cell using ANNs. The training data was generated using the single diode model, and to address the issue of the high variance [58] in the model, a Bayesian regularizer [59] was used by the authors. Results show that the model has high prediction accuracy. A sensitivity analysis (Changing a single parameter at a time and observing its effect on the model output) for the effect of variation in model parameters on the current–voltage curve was also performed. The authors have cited previous works of the literature to justify the use of two hidden layers [60] in their ANN architecture but this can be investigated further for the application of ANNs in solar cell parameter predictions. One point that should be pointed out is that out of the five input parameters used for prediction, two sets of parameters are highly correlated: the maximum power point (P_{MPP}), current at maximum power point (I_{MPP}), and voltage at maximum power point (V_{MPP}). As P_{MPP} is a direct product of I_{MPP} and V_{MPP}, by adding the power, authors indirectly add additional nonlinearity to the model. Nevertheless, the effect of using the current and the voltage only is not reported.

For fitting parameters in a two diode model, Dali et al. [31] applied a genetic algorithm (GA) [61] and particle swarm optimization (PSO) [62]. The training data was produced from the two diode model equation and the squared error cost function [63] was used to minimize learning model hyperparameters [64]. The authors reported that both algorithms had high accuracy but PSO had a faster convergence.

Louzazni et al. [32] also proposed a methodology to predict electrical parameters for an amorphous silicon solar cell model using the firefly algorithm (FA) [65]. Eight model parameters were extracted: photo-current (I_L), the thickness of intrinsic layer (d_i), average mobility-lifetime product for election and hole (μ_τ), built-in field voltage over the i-layer (V_{bi}), diode reverse saturation current (I_S), diode ideality factor N, the series and the parallel resistances (R_S, R_{SH}). The authors reported that their methodology was validated from experimental data and the accuracy was better than other evolutionary algorithms [66]. Although it was reported that this type of solar cells are more difficult to model than a crystalline silicon (c-Si) cell. Future research work can potentially compare this work with conventional Ml techniques that can potentially improve the prediction accuracy further.

III. Application of Ml Techniques Implemented for Process Optimization

Fabrication of solar cells requires multiple processing steps, to achieve the best theoretically estimated device performance, each of these processing steps in a fabrication should be optimized. Similar to device design and architecture optimization, process optimization conventionally can be performed via experimental approaches or process simulations.

Process control and optimization is a complex task given the vast number of chemical and physical mechanisms governing the outcome of that processing step. Li et al. [67] demonstrated optimization of Boron tribromide tube diffusion processes in silicon. A process simulation model was designed first and then calibrated and verified with experimental data. This work provided a meticulous analysis of the different factors (for example, the effect of oxygen and dopant concentration on diffusion kinetics) affecting the diffusion process. Although this work provides an accurate way to model the process, it is complex and resource intensive to replicate for other use cases, given the vast amount of manual tuning that was required to achieve a calibrated model. This makes the model too complex to be reused in a conventional way, but instead it can be abstracted

TABLE II
SUMMARY OF SECTION III

Prediction domain	Article	Cell technology	Target	Model used	Dataset availability	Features	Critical parameter or outcome
Individual process	[69]	PERC	$POCl_3$ diffusion process	GPR	TCAD simulation used to generate data	Diffusion step and oxidation step time and temperatures	Phosphorous profile control parameters can be predicted
	[74]	Silicon ingots	Cz growth O_i	ANN	Raw data not provided	43 (fixed, process and monitored parameters)	Prediction accuracy is effected mostly by monitored parameters
	[76]	SHJ cells	HPT process passivation quality	BO	Available on request from authors	$H2$ pressure and flow rate, RF power, and electrode distance, τ_{eff}	BO is faster than random search for optimized parameters
	[77]	Silicon ingots	Lapping process for uniform thickness	ANN	Raw data not provided	Rotational speed, lapping pressures, and duration	Lapping process control to achieve consistent thickness using ML predictions
Fabrication recipe	[78]	Al-BSF cells	η (process variations)	RF, SVM, AB, ANN	Source code and data are available publically	47 parameters from 10 processing steps	ANN has highest reported accuracy
	[79]	PERC	η variation by tool throughput	Statistical modeling	TCAD simulations	Time (Production)	Root cause analysis for efficiency variations modeled
	[80]	p type Cz Si cells	Hourly trend in η	Decision trees	Research data are not shared.	329 (13 clusters)	Impact of different features on temporal performance
	[81]	p type mc Si cells	Hourly trend in η variation in production line	GPR,Monte-carlo simulations	TCAD Simulation	Rear passivation quality, bulk doping, texturing, front metal shading	3 material parameters responsible for 80% efficiency variations were identified

to train an Ml model that can predict the model performance with an acceptable accuracy. One hypothesis can be to treat the model as a black box and statistically vary only six parameters (deposition temperature, drive-in temperature, postoxidation temperature, deposition duration, drive-in duration, and postoxidation duration) and observe the resulting diffusion profile. This combination of six parameters can act as input features for an ML model and the resulting diffusion profile can be the prediction target. A trained ML model can then accurately interpolate and extrapolate the result of a complex numerical simulation model providing exponentially faster prototyping time.

This section explores the various efforts in the literature that implemented data science techniques to optimize either a single processing step or an entire fabrication recipe. A summary of this section is provided in Table II.

A. Application of ML to Improve the Individual Processing Steps

Emitter diffusion profiles have a direct impact on the final device performance [68]. Wagner-Mohnsen et al. [69] used ML techniques and GA to optimize the $POCl_3$ diffusion processing in PERC solar cells. Training data was generated by running 2400 Sentaurus simulations [70], including both process and device simulations. A GPR model with rational quadratic kernel [71] was used as the ML algorithm. To find the required process parameters to achieve the highest power conversion efficiencies, mutation, and tournament selection algorithm [72] were implemented. Results show that process parameters to achieve a phosphorus diffusion profile can be estimated using the ML and GA techniques. This work provides a way to optimize individual process parameters required to achieve a certain device efficiency, future research can extend this work by emphasizing on the correlation between individual process parameter variations and final device efficiency.

Interstitial oxygen in a solar cell material is a major factor for degradation of minority carrier lifetime [73], evaluation and control of this impurity concentration during crystal growth therefore can be one way to optimize the final device performance. Kutsukake et al. [74] presented a way to predict interstitial oxygen concentration (O_i) in Czochralski (Cz) grown silicon crystal using neural networks. Training data was collected from 450 ingots grown in the same furnace and the O_i concentration was measured using Fourier transform infrared spectroscopy. From an initial set of 100 features, the authors did a correlation analysis and reduced the features to 43. The model was a fully connected feed-forward NN [75] with 3 layers and 47 nodes in each layer and the model was trained for 20 000 epochs. Results show that the trained model can make predictions for test data with high coefficient of determination (R^2 score). This work provides a thorough discussion on the possibility of use of Ml algorithms in process control where conventional simulation and modeling techniques cannot be efficient. For example, certain variations inside the crucible cannot be modeled in a traditional way. The authors have also classified input features into the following three categories: fixed parameters, process parameters, and monitored parameters. This classification narrows the domain of control over the crystal growth. The possibility to improve upon this current model is also discussed by proposing use of past data from time series of an ingot growth. This work is one of a kind and can be a fundamental block for approaching control of other processes using data science techniques.

SHJ solar cell power conversion efficiency is significantly affected by the carrier selective contacts. The main objective of these contacts is to efficient charge separation and to reduce carrier recombination. Miyagawa et al. [76] have presented a data science-based approach to optimize the hydrogen plasma treatment (HPT) for improving the passivation quality of heterostructure between titanium oxide and c-Si. Optimization of the HPT process conditions is required to achieve the best passivation quality. In this work, the authors have used Bayesian optimization (BO) rather than relying on conventional random or grid search for parameters. The six parameters (or ML features) that are being optimized are process temperature, process time, hydrogen ($H2$) pressure and flow rate, radio frequency (RF) power, and electrode distance. Effective carrier lifetime (τ_{eff}) was the parameter used to qualitatively assess the passivation quality. Using the upper confidence bound of the uncertainty of a GPR model, the acquisition function [47] of BO was defined. The training data was generated from 10 samples, on which HPT treatment was performed. Using the exploration–exploitation approach [47] required experimentation points were determined. Results show that this approach inferred the optimized values of the six parameters much faster than the random search approach. The authors have also explained in detail their findings and their correlation with device physics. This work presents a very efficient way of process optimization, one which minimizes redundant experimentation and computational resources, while also providing a degree of uncertainty for each prediction (as the underlying methodology for acquisition function is GPR, a probabilistic learning methodology).

Silicon ingots grown from Cz process are conventionally cut into wafers using a diamond wafer saw, but this cutting methodology creates irregularities on the wafer surfaces that lead to defect formation. The lapping process is used, therefore, to remove the thickness variations after cutting the ingots into wafers, this helps in maintaining a flat surface on the wafer as well as to minimize any necessity of polishing later. Surface roughness (Ra) can be used to qualitatively assess the lapping process. An abrasive slurry mixture like aluminium oxide ($Al2O3$) is used in the lapping process, and long operations using this slurry mixture can be expensive, which is one key disadvantage of the lapping process. Ozturk et al. [77] presented a way to reduce the extent of this inherent disadvantage by optimizing the process parameters associated with lapping: abrasive slurry flow rate, the particle size of abrasive material used in slurry, application pressure, the rotation speed of wafer on cast iron plate. The authors also gave an insight into an existing literature where non-Ml-based methodologies are applied to address this optimization and pointed out that the number of studies in this domain are very limited. To train an ANN model 218 different data points were measured, rotation speeds, lapping duration, and lapping pressures were model input and measured Ra values were the model output. The neural network used was a feed-forward fully connected network with three layers and ten neurons in the middle hidden layer. This network was trained for 1000 epochs with a learning rate of 0.01. Using this ANN prediction the authors were able to confirm a few theories about the lapping process, for example, extending lapping duration over 35 min in all rotation speeds has no significant advantage. This work highlights the core advantages of including data science techniques in process engineering, mainly the use of existing data future results can be optimized with minimal disruption of the entire fabrication process.

B. Application of Ml to Improve a Fabrication Recipe

Instead of focusing on a single process step, multiple processes, or the entire fabrication recipe can be optimized, using conventional modeling or experimental approaches, the number of variables (process conditions and/or mathematical variables in case of analytical/numerical modeling) involved in optimization can exponentially increase making the process time and resource expensive. Data science-based approaches can be useful in such cases as learning on a large number of variables is feasible. Buratti et al. [78] have addressed the issue of natural variations in production lines and proposed an ML-based approach to optimize multiple processes simultaneously. Another goal that the authors had was to keep improving the efficiency to cost ratio. The authors used several ML methods: SVR, RFs [82], adaptive boosting (AB) [83], 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. The demonstration was done on Al-BSF solar cell production line. A total of 47 process parameters from ten processing steps were used as the features of the ML models, and a five fold cross validation [84] was used to test the model accuracy. The optimized process parameters obtained from GA were further tested using the PC1D simulation tool. RMSE [85] and R^2 score were used to quantify the model accuracy, and it was reported by the authors that NN provided the best accuracies for larger datasets (with more than 1000 training data points). This work provides a meta-heuristic approach [86] to address the issue of multiprocess optimization of solar cell production lines without increasing the complexity exponentially. This work can be extended in future research by exploring different implementation strategies of the ML model, for example, tuning the structure of the ANN architecture.

To understand the effect of a fabrication tool throughput on the final cell efficiency, Altermatt et al. [79] proposed a statistics-based approach, where the performance of every individual tool involved in the fabrication of 552 000 PERC solar cells in a production line were investigated over 2 weeks. The efficiency distribution of the produced cells was also recorded. Then, a technology computer aided design (TCAD) simulation for these cells was used to achieve the baseline measured efficiency by statistically varying the input parameters of the simulation model. The effect of individual processing steps was inferred on the final device efficiency. Modeling of semiconductor devices involves the description of the transport of charge carriers under the influence of potential distributions, properties of semiconductor material, and doping distributions. These basic equations are Poison, current continuity, and transport equations. These are partial differential equations (PDE) in nature,which are coupled together, and they are defined in a n dimensional space (1-3), an analytical solution for these equations is not possible, so they are solved by numerical methods which have high-time complexity attached. An Ml model can predict the results of these simulation models as the inference is different compared to solving PDE's. For example, if a three layer trained neural network model is used for regression, it starts by creating linear functions at the first layer (between different connected nodes), the second layer basically combines these multiple linear functions into multiple nonlinear functions. The third layer combines these further to create further nonlinear functions. The inference is faster as the weights associated with each node (variable of these nonlinear functions) are precalculated during the training stage.

Wasmer et al. [80] introduced an ML-based approach to solving the same problem. The author's goal was to understand the global hourly time trends of solar cell efficiency seen in a 1-week dataset. An ensemble of decision trees [87] with feature subsampling [88] was the ML model used in this work. The training data was collected from 500 000 solar cells using inline measurements over a period of 1 week. The total number of features was 329, and the authors clustered these large sets of features in groups with sizes varying from 1 to 13. The reasoning given was that many of these features might be highly correlated. Although instead of using a technique like principal component analysis (PCA) [89], the authors proposed a hierarchical clustering technique [90] to group the features. Their reasoning to do this was that in PCA it will be unclear, which features were joined. In the proposed hierarchical clustering techniques, features were grouped based on their similarity with respect to their impact on a cell performance metric. The designed Ml model was used to perform temporal predictions (time as one of the features) of the solar cell efficiency. It was proposed that, as the prediction accuracy was very high, this methodology could be implemented for actual production lines. A unique and innovative aspect of this work is that the focus is not to achieve an Ml model with high prediction accuracy. Instead, the focus is to use ML as a tool to understand the impact of different features of solar cell production on the temporal evolution of produced solar cell performance.

Another approach to model the power conversion efficiencies from a production line using meta-modeling (the author refers to Ml models that can predict simulation model results) and Monte Carlo simulations [91] is presented in Wasmer and Greulich [81]. In this work, first in line measurements and device characterization are done to collect the measurement database, then a numerical simulation model for the solar cell is designed, next a GPR Ml model is created to interpolate the simulation results (i.e., efficiency), after that, using Monte Carlo approximation input parameter set in a range is created and finally a sensitivity analysis is performed for these input parameters. The authors were able to identify three material parameters that are responsible for 80% of the variations in the efficiency and propose that if these parameters are optimized the baseline efficiency will improve. This work provides an efficient way to utilize the massive measurements database created during solar cell production by combining characterization techniques, numerical simulations, probabilistic modeling, and sensitivity analysis to improve upon the baseline efficiency of a given production line.

IV. Application of ML Techniques Implemented for Surface Defects (Cracks And Microcracks Using Luminescence Imaging) and Material Bulk Defects

One major bottleneck for any mass-produced silicon solar cell technology to achieve and sustain peak power conversion efficiencies are defects. These defects can occur during the manufacturing itself or they can occur during the device operation. There are also multiple mechanisms that can introduce defects in a solar cell device at different stages of the lifetime of the solar cell, for example, defects in the bulk are mainly introduced during the material growth while cracks and microcracks can be introduced in the device during and after manufacturing.

There are conventional methodologies illustrated in the literature to identify, avoid, and mitigate defects in a solar cell like the work presented by Michl et al. [92], where they investigated different loss mechanisms in solar cells. This work demonstrates that for investigating a single loss mechanism, for example, surface recombination, multiple data sources can be used like PL images or surface recombination velocity values, and these different sources are correlated to an extent. This provides a possibility to combine these different sources of learning via an ML model. In this section, our focus is on those efforts, which have used ML-based methodologies to identify and rectify defects on the surface and in the bulk. A summary of this section is provided in Table III.

A. *Learning Methodologies to Investigate Surface Defects*

The implementation of a CNN [104], [105] based classifier to detect surface defects is presented in detail by Akram et al. [93], where the authors have used electroluminescence (EL) imaging to extract features (EL images with and without defects). They cited the advantages of EL imaging in distinguishing cracks/microcracks within a solar cell. Previous similar attempts in the literature using Ml techniques and luminescence imaging are also discussed and the need for their own implementation is given. The authors have also discussed the advantages of CNN over a fully connected NN in detail for image-based learning. For training a CNN first, the EL images were annotated manually for different kinds of defects (materials, cracks, finger defects, and others). Then, different CNNs were tested and finally, their classification performance was evaluated against a test dataset. Different VGG-based [106] deep architectures were tested and it was found that a six-layer network (that includes two fully connected layers) performs the best. It was also reported that the addition of batch normalization and regularization helped in reducing overfitting. The images used for training and testing were of resolution 100 × 100 and were grey scaled. Their result showed that the CNN network misclassified background as defects for polycrystalline solar cells and in mono crystalline solar cells minor defects were misclassified. Emphasis was given during this research to keeping the model lightweight (in terms of the computational requirements). This is advantageous as the authors claimed that the model can run on lower end consumer grade computing equipment. This is ideal for an in line industrial inspection use case.

Defects in polycrystalline solar cells are difficult to investigate compared to mono-c-Si solar cells as crystallographic anomalies can be falsely classified as cracks/microcracks. Zhang et al. [94] proposed a way to counter this issue by using transfer learning [107]. In their work, a model that was trained for labeled mono-crystalline EL images was used as the base model for inference on unlabeled polycrystalline solar cell EL images. To

TABLE III
SUMMARY OF SECTION IV

Prediction domain	**Article**	**Cell technology**	**Target**	**Model used**	**Dataset availability**	**Features**	**Critical parameter or outcome**
Surface defects	[93]	Mono, multi crystalline solar cells	Crack, finger, and material defects	VGG	Publicly available	Manually labeled EL images	Accuracy of 93.02%, Scalable on low end hardware
	[94]	Poly crystalline solar cell	Cracks and microcracks	DAN	Not publicly accessible	EL images labeled with defect probability	Transferred learning from mono to poly crystalline solar cell
	[95]	Mono, multi crystalline solar cells	Finger interruptions, material defect, inter-connection degradation, cracks & insulation faults	SVM & CNN	Publicly available	Module EL images (Individual cells labeled)	Scalability of CNN & SVM from mobile to high performance hardware
	[96]	Polycrystalline solar cell	Defect regions (boundary boxes)	CNN with RPN (U-Net)	Not publicly accessible	Gray scale images	Boundary boxes for defect regions with RPN implementation
	[97]	PERC	Empirical digital twin for manual defect classification	DenseNet	Research data are not shared	EL images, IR images, and reflectance values	Issue of erroneous data labeling in luminescence images is solved
	[98]	Mono, multi crystalline solar cells	micro and large scale cracks	Deep Siamese CNN	Public access provided in separate article	EL images (Divided symmetrically at center half)	This model can work on nonuniformly captured images
	[99]	Multi-crystalline solar cells	Finger interruption, cracks and black core defects	CAN	Publicly available	EL images	Ability to filter out background information
	[100]	Silicon wafers (mc- Si)	Pixel level mapping of V_{OC}	RAM CNN-based regression	Not publicly accessible	PL images	Prediction of cell performance at bare wafer level
	[101]	PERC	FF, V_{OC}, J_{SC}, η	Deep CNN, AB, RF	No raw data provided	PL images	Model prediction is robust image manipulations
	[102]	No specific cell technology	η bins		Not publicly available	EL images	Model can work on both full and half cut cells
Bulk defects	[103], [17]	Not specific to one technology	σ_t, σ_p, N_t, N_{dop}, E_t	ANN, RF, SVM, ada, and grad boosting	Publicly available	Carrier lifetime profiles	Model was robust even when noise was added to input data

reduce the loss in learning when the model is transferred, and to avoid model overfitting on mono-crystalline solar cell images, the authors have used a deep adaptation network (DAN) [108] using the multiple kernel variant of maximum mean discrepancies [109]. The test dataset contained 2886 mono-crystalline cells and 2154 polycrystalline cell grey-scale EL images with a resolution of 300×300 pixels. Results show that the prediction accuracy for polycrystalline cells was around 77%, and the authors acknowledged the need to perform better data preprocessing (to remove data defects and mislabeled data). This work lays the groundwork for domain adaptation and transfer learning from source to target tasks in PVs.

Deitsch et al. [95] explored two approaches for defect detection in solar cells using images captured at the module level. Those two approaches were explored different learning algorithms based on the degree to which they can utilize hardware-based acceleration for training and inference of the model. SVMs were trained on the features of the EL images, while CNN used information of individual image pixel for a cell. For training an SVM model, luminescence images of a module were segmented at individual cell level and local descriptors were extracted, which were combined then to compute a global representation, which can classify a cell as defective or functioning. Two kernels were used, namely linear and RBF [40]. The concept of masking was introduced to separate background information of the module (like bus bars). An VGG-19 network was modified to design a deep CNN, with a single neural at the last layer providing defect probability. The dataset for training was composed of 1968 samples. Results show that the CNN performs better than the SVM by a margin of 6%. Although the difference in prediction accuracy is not significant, the key points to take from the analysis presented are the time complexities associated with the two learning algorithms. Mainly, the SVM can efficiently utilize and scale to a diverse set of hardware devices, taking significantly less time for model training. Although CNN's can provide faster inference. SVM models are CPU intensive while CNN's can perform best on a GPU. This result can help future result to establish a tradeoff between unique use cases, available hardware resource and training and inference time budget available.

Han et al. [96] proposed segmentation of defects in polycrystalline solar cells using CNNs. They have addressed the issue of information content in industrial luminescence imaging (as compared to natural images, which have clear semantic boundaries, industrial images are usually in grey-scale and lack semantic boundaries). They have designed a defect region proposal mechanism using region proposal networks (RPN) (see, e.g., [110]), where an RPN provides binary classification as well as regression of boundary boxes around the defect regions. The dataset contained 106 defective images with a resolution

1024 × 1024. The authors also experimented by changing the patch size of images for training and found that the highest patch size (256 × 256) had the most optimal results. A modified version of U-Net architecture was used in this work and when compared against the base U-Net model, FCN-8 s, and DeepLabv2, it was found that the proposed methodology outperformed these methods.

Kunze et al. [97] presented an interesting idea of addressing the issue related to data annotations and labeling in ML models that learn from luminescence images of solar cells. These issues range from erroneous data labeling and the time complexity associated with manual annotations. They have presented concept of an empirical digital twin (EDT), which require minimal human intervention. The initial training was done on reflectance values of different solar samples and their EL and infrared images. Using a regression layer attached at the end of a CNN, cell quality parameters like cell V_{oc} can be established. An argument was made to support the fact that the EDT has already learnt different defect classes and a minimal dataset for a particular defect type can be used to train a subset of EDT models using active learning (with human intervention). This training will be performed iteratively (training data is divided into batches to reduce the error in training prediction). The training dataset showcased consisted of 1600 c-Si PERC solar cells. The DenseNet [111] base network was modified for this use case. A ground true value (defect information) was attached with each individual sample and the precision, recall, and F1 scores were established using true positive, false negative, and false positive classification of defects. Results show that the proposed learning approach has better prediction accuracy than conventional supervised learning approaches, with F1 score for defect class finger interruption of more than 0.99.

A novel way of improving the prediction accuracy of defect classification by learning the left and right half of an EL image was presented by Acharya et al. [98], where they have used deep Siamese CNN's [112]. As solar cells are symmetric across their surface, the authors proposed using left and right half of the EL image as two separate inputs to the two branches of the Siamese CNN. Prior to feeding the EL images to the CNN, they were preprocessed (geometrically transformed). This article explained the model implementation in detail. As the goal of this work was to predict four different kinds of defects (microcrack defects, large-scale defects, defects with low resolution, and no defects), a multiclassification was performed by the neural network. A comparison between the results of this work, i.e., accuracy of confusion matrix, the area under curve and that of earlier published results was also performed. This work can be helpful for industrial use cases, especially for field inspection of solar modules, where EL images may not be uniformly captured (for example, they may have different light exposure on the left and right sides of the images).

Another innovative approach to detect defects more accurately by suppressing background noise in an EL image using a complementary attention network (CAN) [113] was presented by Su et al. [99]. The authors have explained the capability of channel-wise attention to discard background information. Particularly, this is used to justify the use of CAN with RPNs.

Demant et al. [100] investigated deep CNNs and PL imaging for regression analysis, they were predicting the electrical performance of the solar cell, and they also introduced mapping of open-circuit voltage (V_{oc}) using regression activation mapping (RAM) [114]. PL images of 3000 wafers were used to train a CNN, and testing was done on wafers from different silicon bricks, it is claimed that as the testing is done on data that the model has not seen during the training stage and still performs efficiently (MAE was the metric for checking the prediction quality), the model can be generalized. Results show that inference using the model is done in milliseconds, efficiency, and V_{oc} are predicted with lower MAE, while prediction of short-circuit current density (J_{sc}) is less accurate. This article provides an innovative way to forecast the performance of a solar cell manufactured from a given wafer, mapping of V_{oc} over the wafer using RAM is also helpful to understand the wafer quality, and the authors have also compared it to j_0 images for verification. Measures to improve the model are also proposed, mainly increasing the resolution of the RAM mapping of V_{oc} and by additional feature vectors like brick lifetime.

Another end-to-end prediction model was proposed by Demant et al. [101], which utilizes a CNN-based regression model to predict electrical parameters—efficiency, open-circuit voltage, short-circuit current, and fill factor (FF) from PERC solar cell PL images. The authors have focused on increasing the generalization capabilities of their model by curating material data of 7300 wafers sourced from 10 different manufacturers. In order to augment the training data [115] for further model robustness improvement, the images were randomly cropped at both borders, flipped, and rotated. To assess the model generalization and check for possible biases, testing was done for different scenarios, where the authors tested the prediction quality of the model for unknown bricks and unknown manufacturers (in other words, data that the model has not seen during its training and validation stages). Results show that the highest prediction accuracy was achieved for the prediction of V_{oc}, followed by η or J_{sc}, and, FF. The authors have acknowledged the limitation of the model caused by insufficient information captured by PL images (when multiple surface defects are present). They proposed that the inclusion of additional measurement data can complement the visual data in PL images and accuracies can be further improved.

Buratti et al. [102] presented an interesting use case of end-to-end prediction modeling with regression-based final output using deep CNN's. The goal of this proposed work was to predict and bin solar cell efficiencies. A total of 30 000 full cells were used to capture EL images and the corresponding electrical parameters were measured for training and validating the model. A 0.2% efficiency bin size was chosen. The authors tested the following five different CNN architectures: AlexNet [116], ResNet [117], DenseNet, VGG, and SqueezeNet apart from a custom-built CNN architecture. For binning (regression) also, multiple ML algorithms were tested: RF [118], AdaBoost, gradient boost [119], and ANN. The model were tested on full size cells as well as half cut cells. Results showed that the custom built CNN architecture was the best performing algorithm, and AdaBoost was the best algorithm (with R square score of above

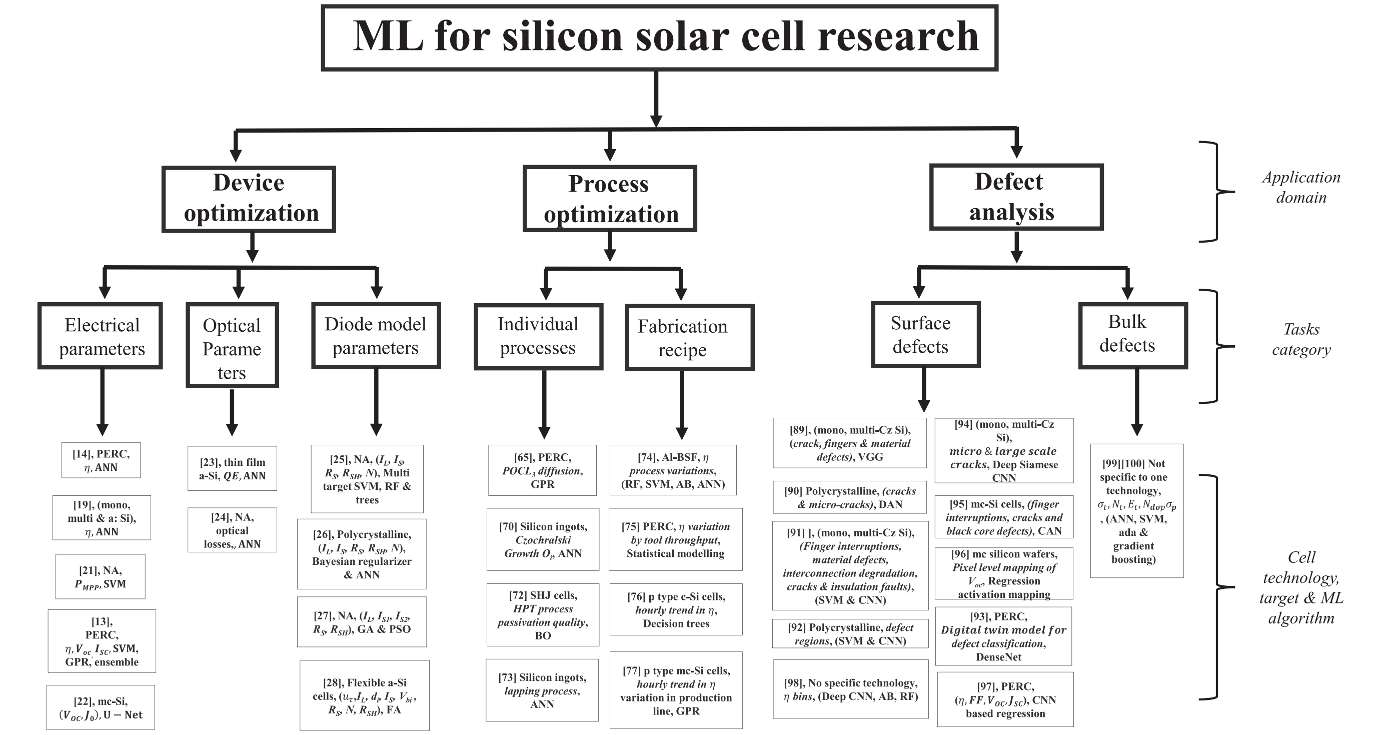
ML for silicon solar cell research
Device optimization
Process optimization
Defect analysis
Application domain
Electrical parameters
Optical Parameters
Diode model parameters
Individual processes
Fabrication recipe
Surface defects
Bulk defects
Tasks category
[14], PERC, η, ANN
[19], (mono, multi & a: Si), η, ANN
[21], NA, P_{MPP}, SVM
[13], PERC, η, V_{oc}, I_{SC}, SVM, GPR, ensemble
[22], mc-Si, (V_{OC}, J_0), U − Net
[23], thin film a-Si, QE, ANN
[24], NA, optical losses,, ANN
[25], NA, (I_L, I_S, R_S, R_{SH}, N), Multi target SVM, RF & trees
[26], Polycrystalline, (I_L, I_S, R_S, R_{SH}, N), Bayesian regularizer & ANN
[27], NA, (I_L, I_{S1}, I_{S2}, R_S, R_{SH}), GA & PSO
[28], Flexible a-Si cells, (u_τ, I_L, d_i, I_S, V_{bi}, R_S, N, R_{SH}), FA
[65], PERC, $POCL_3$ diffusion, GPR
[70] Silicon ingots, Czochralski Growth O_i, ANN
[72] SHJ cells, HPT process passivation quality, BO
[73] Silicon ingots, lapping process, ANN
[74], Al-BSF, η process variations, (RF, SVM, AB, ANN)
[75] PERC, η variation by tool throughput, Statistical modelling
[76] p type c-Si cells, hourly trend in η, Decision trees
[77] p type mc-Si cells, hourly trend in η variation in production line, GPR
[89], (mono, multi-Cz Si), (crack, fingers & material defects), VGG
[90] Polycrystalline, (cracks & micro-cracks), DAN
[91]], (mono, multi-Cz Si), (Finger interruptions, material defects, interconnection degradation, cracks & insulation faults), (SVM & CNN)
[92] Polycrystalline, defect regions, (SVM & CNN)
[98], No specific technology, η bins, (Deep CNN, AB, RF)
[94] (mono, multi-Cz Si), micro & large scale cracks, Deep Siamese CNN
[95] mc-Si cells, (finger interruptions, cracks and black core defects), CAN
[96] mc silicon wafers, Pixel level mapping of V_{oc}, Regression activation mapping
[93], PERC, Digital twin model for defect classification, DenseNet
[97], PERC, (η, FF, V_{OC}, J_{SC}), CNN based regression
[99][100] Not specific to one technology, $\sigma_t, N_t, E_t, N_{dop}\sigma_p$, (ANN, SVM, ada & gradient boosting)
Cell technology, target & ML algorithm

Fig. 2. Hierarchy of data science-based silicon solar cell research (NA: Not available).

99.8%) for binning. It was also shown that as the bin sizes are extremely small, the accuracies were affected by it, and many incorrect binning was done only in the neighboring bins. This work lays can be used as a fundamental module to build binning algorithms for cell quality inspection in an industrial use-case.

B. Learning Methodologies to Investigate Material Bulk Defects

The power conversion efficiency of a solar cell is also affected by bulk defects, Buratti et al. [103] presented a regression-based strategy to extract defect parameters that directly affect the temperature- and injection-dependent lifetime spectroscopy. The authors simulated more than 100 000 by varying the different parameters: Defect energy level (E_t), Capture-cross sections (σ_t, σ_p), defect density (N_t), doping density (N_{dop}), excess carrier concentration (Δn), and temperature (T), these parameters affect the lifetime spectroscopy. Then, 500 features were used to design a regression model that is targeted to predict E_t, σ_t, and σ_p. The authors decided to separate the training of the ML model for defect energies that are in the upper band gap from the ones in the lower band gaps. The regression model was based on an ensemble algorithm: RFs. The coefficient of determination score was used to judge the prediction quality and it was more than 0.96 for all the prediction models. The findings also establish that the ML algorithm was able to predict defect parameters without prior knowledge of the Shockley-read-hall (SRH) equation.

Buratti et al. [17] described their idea further in another article, where different Ml models (RF, AB, gradient boosting, NN, and SVM) were compared, and it was shown that all the ML models were able to back predict parameters like E_t without explicit definition physics-based models. An interesting experiment was presented where random Gaussian noise was added to SRH lifetime profiles and it was shown that the ML models were able to predict defect parameters with accuracy not dropping. Simulated data generation and Ml training methodology was presented along with directions to recreate this work in detail. This research provides a novel alternative approach to detect bulk defects compared to traditional characterization approaches and prove that Ml models are even robust to noise in input data, which is promising factor to extend this concept to other directions. As in most cases experimental data is noisy, an ML model trained on simulated data can be used for inference over this noisy data.

V. Conclusion

This article presents a comprehensive overview of data science methodologies applied to PV research in its three main branches, which are device optimization, process optimization, and defect analysis on surfaces and inside the bulk of solar cells. A hierarchy of this review is shown in Fig. 2.

A common trend amongst the reviewed articles is that data science-based approaches, such as Ml and meta-heuristic algorithms can potentially accelerate research in the PV domain. This improvement can be in terms of time, as multiple conceptual device architectures and process recipes can be prototypes much more rapidly than conventional experimental approaches, defect analysis is also accelerated using learning approaches, such as computer vision and DL on images. In addition to faster research output, ML approaches also significantly cut down the resources needed in a research. These resources can range from the person/hours invested to the computational to the material resources required.

The number of published efforts in the domain of applied ML in PV research is increasing over the years. These publications are the foundations on which future explorations in the domain of applied data science-based PVs research will be performed.

Frequently, the detailed description of the devices, fabrication processes, or simply the methodologies for acquiring the necessary data leaves low room for the information necessary to recreate the research from the point of view of the artificial intelligence applied in these works. Indeed, in order to fully describe any Ml methodology, it needs to be broken into the used structure, the chosen optimization criterion, and the particular algorithm that implements said criterion over the structure. The structures need to be fully described for the experiments to be repeated. For example, an NN or any deep structure as a CNN has a variety of structural parameters fixed before the experiments are run, such as the number of layers, the dimensions of the kernels, and so on. In many cases, the training criterion is implicitly understood when the type of technique is mentioned, as, for example, in GP (where maximum a posterior is used) or SM (that uses maximum margin), but in other situations, the criteria are multiple. This can also be said about the algorithms used for the optimization. Besides, these algorithms have parameters that need to be optimized and intrinsic quantities and procedures that have to be mentioned, as for example, the fractions of data used for training, validation and test and others. A detailed description of all the above is of paramount importance for proper reproduction of the presented procedures and to boost the already increasing efficiency of data science-based approaches even further.

Another oversight observed in many reviewed articles is an effort to make the implemented ML model more generalized. A common issue with training ML models is the bias-variance tradeoff, when the model has a high bias toward the training data, adding more training data will not improve the model prediction performance for test data, instead of increasing model accuracy, decreasing regularization, or boosting can help in reducing the bias. When the model has high variance, increasing training data, reducing the model complexity, and increasing regularization or bagging can help. In order to ensure that an ML model implementation has good generalization properties, it must be evaluated. This can even help in transferring the learning from a trained Ml model to another research work, without coming up with a novel model architecture.

With the increasing accessibility of cloud-based computing, computational capacity of modern CPU's and GPU's and increasing efficiency of software frameworks, there is a very high probability that the restrictions in Ml tasks with respect to memory and compute capacity as of now will be removed in the near future. This will be particularly helpful in DL and computer vision algorithms for image-based learning. These improvements will result in faster introductions of novel research ideas, as well as lead to increased implementations in a mass fabrication industrial environment as real-time Ml predictions will allow in-line characterization of cells and modules.

Possibilities of introduction of newer methods of using learning methods in PVs research are inherently dependent on this work being done in this domain, as these current literary works act like fundamental building blocks for further improvements. Therefore, a comprehensive introduction to the methodologies that have been implemented in their research work by authors is a fundamental necessity.

References

[1] M. A. Green, “The passivated emitter and rear cell (PERC): From conception to mass production,” *Sol. Energy Mater. Sol. Cells*, vol. 143, pp. 190–197, 2015.

[2] Y. Chen et al., “From laboratory to production: Learning models of efficiency and manufacturing cost of industrial crystalline silicon and thin-film photovoltaic technologies,” *IEEE J. Photovolt.*, vol. 8, no. 6, pp. 1531–1538, Nov. 2018.

[3] F. Tao, Q. Qi, A. Liu, and A. Kusiak, “Data-driven smart manufacturing,” *J. Manuf. Syst.*, vol. 48, pp. 157–169, 2018.

[4] H. Zhu et al., “Design investigation on 100 μm-thickness thin silicon PERC solar cells with assistance of machine learning,” *Mater. Sci. Semicond. Process.*, vol. 137, 2022, Art. no. 106198.

[5] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, http://www.deeplearningbook.org

[6] J. Bergstra et al., “Theano: Deep learning on GPUS with python,” in *Proc. NIPS BigLearning Workshop*, 2011, vol. 3, pp. 1–48.

[7] S. Raschka and V. Mirjalili, *Python Machine Learning: Machine Learning and Deep Learning With Python, Scikit-Learn, and TensorFlow 2*. Birmingham, U.K.:Packt Publishing Ltd., 2019.

[8] J. Brownlee, *Deep Learning With Python: Develop Deep Learning Models on Theano and TensorFlow Using Keras*. San Francisco, CA, USA: Machine Learning Mastery, 2016.

[9] E. Stevens, L. Antiga, and T. Viehmann, *Deep Learning With PyTorch*. Shelter Island, NY, USA: Manning Publications, 2020.

[10] G. Ke et al., “Lightgbm: A highly efficient gradient boosting decision tree,” in *Advances in Neural Information Processing Systems*, I. Guyon et al., Eds., vol. 30. Red Hook, NY, USA:Curran Assoc., Inc., 2017.

[11] T. Chen and C. Guestrin, “XGBoost,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 785–794.

[12] G. Varoquaux et al., “Scikit-learn: Machine learning without learning the machinery,” *GetMobile: Mobile Comput. Commun.*, vol. 19, no. 1, pp. 29–33, Jun. 2015.

[13] N. Chen and A. Ebong, “Towards 20% efficient industrial AL-BSF silicon solar cell with multiple busbars and fine gridlines,” *Sol. Energy Mater. Sol. Cells*, vol. 146, pp. 107–113, 2016.

[14] Y. Tsunomura et al., “Twenty-two percent efficiency hit solar cell,” *Sol. Energy Mater. Sol. Cells*, vol. 93, no. 6, pp. 670–673, 2009.

[15] A. Blakers, N. Zin, K. McIntosh, and K. C. Fong, “High efficiency silicon solar cells,” *Energy Procedia*, vol. 33, pp. 1–10, 2013.

[16] H. Wagner-Mohnsen and P. P. Altermatt, “A combined numerical modeling and machine learning approach for optimization of mass-produced industrial solar cells,” *IEEE J. Photovolt.*, vol. 10, no. 5, pp. 1441–1447, Sep. 2020.

[17] Y. Buratti, Q. T. Le Gia, J. Dick, Y. Zhu, and Z. Hameiri, “Extracting bulk defect parameters in silicon wafers using machine learning models,” *NPJ Comput. Mater.*, vol. 6, no. 1, 2020, Art. no. 142.

[18] “International technology roadmap for photovoltaic 2020 results,” pp. 30–31, 2021. [Online]. Available: https://www.vdma.org/international-technology-roadmap-photovoltaic

[19] G. S. Thirunavukkarasu et al., “Optimization of mono-crystalline silicon solar cell devices using PC1D simulation,” *Energies*, vol. 14, no. 16, 2021, Art. no. 4986.

[20] D. Clugston and P. Basore, “PC1D version 5: 32-bit solar cell modeling on personal computers,” in *Proc. IEEE Conf. Rec. 26th Photovolt. Specialists Conf.*, 1997, pp. 207–210.

[21] A. Fell, “A free and fast three-dimensional/two-dimensional solar cell simulator featuring conductive boundary and quasi-neutrality approximations,” *IEEE Trans. Electron Devices*, vol. 60, no. 2, pp. 733–738, Feb. 2013.

[22] S. S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Upper Saddle River, NJ, USA: Pearson Educ., 2009.

[23] 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, 2017, Art. no. e0184561.

[24] M. A. Green, “Crystalline and thin-film silicon solar cells: State of the art and future potential,” *Sol. Energy*, vol. 74, no. 3, pp. 181–192, 2003.

[25] D. Nurwaha, “Comparison of kernel functions of support vector machines: A case study for the solar cell output power prediction,” *Int. J. Energy Appl. Technol.*, vol. 7, no. 1, pp. 1–6, 2020.

[26] A. S. Kovvali, M. Demant, and S. Rein, “Spatially resolved material quality prediction via constrained deep learning,” in *Proc. IEEE 46th Photovolt. Specialists Conf.*, 2019, pp. 3059–3062.

[27] M. Kaya and S. Hajimirza, “Rapid optimization of external quantum efficiency of thin film solar cells using surrogate modeling of absorptivity,” *Sci. Rep.*, vol. 8, no. 1, pp. 1–9, 2018.

[28] M. K. Shivhare, A. Yadav, S. R. Pillai, V. K. Vashishtha, and M. Kumar, “Prediction of anti-reflection coating thickness of a solar cell using artificial neural network,” in *Proc. IEEE 15th Int. Conf. Ind. Inf. Syst.*, 2020, pp. 28–33.

[29] R. Shah, “Solar cell parameters extraction using multi-target regression methods,” in *Proc. IEEE Int. Conf. Environ. Elect. Eng. IEEE Ind. Commercial Power Syst. Europe*, 2020, pp. 1–6.

[30] B. Cortés, R. Tapia Sánchez, and J. J. Flores, “Characterization of a polycrystalline photovoltaic cell using artificial neural networks,” *Sol. Energy*, vol. 196, pp. 157–167, 2020.

[31] A. Dali, A. Bouharchouche, and S. Diaf, “Parameter identification of photovoltaic cell/module using genetic algorithm (GA) and particle swarm optimization (PSO),” in *Proc. IEEE 3rd Int. Conf. Control, Eng. Inf. Technol.*, 2015, pp. 1–6.

[32] M. Louzazni, A. Khouya, A. Crăciunescu, K. Amechnoue, and M. Mussetta, “Modelling and parameters extraction of flexible amorphous silicon solar cell A-SI:H,” *Appl. Sol. Energy*, vol. 56, no. 1, pp. 1–12, 2020.

[33] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, 2004.

[34] B. Schölkopf and A. Smola, *Learning With Kernels. Cambrigde*, Cambridge, MA, USA:MIT Press, 2002.

[35] J. T. Kwok and I. W. Tsang, “Linear dependency between ε and the input noise in ε-support vector regression,” *IEEE Trans. Neural Netw.*, vol. 14, no. 3, pp. 544–553, May 2003.

[36] B. Gu et al., “Accurate on-line -support vector learning,” *Neural Netw.*, vol. 27, pp. 51–59, 2012.

[37] S. Weisberg, *Applied Linear Regression*. Hoboken, NJ, USA:Wiley, 1985.

[38] W.-Y. Loh, “Classification and regression trees,” *WIREs Data Mining Knowl. Discov.*, vol. 1, no. 1, pp. 14–23, 2011.

[39] Y. Ren, L. Zhang, and P. Suganthan, “Ensemble classification and regression-recent developments, applications and future directions [review article],” *IEEE Comput. Intell. Mag.*, vol. 11, no. 1, pp. 41–53, Feb. 2016.

[40] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. Cambridge, MA, USA:MIT Press, 2005.

[41] M. Minsky, *Perceptrons; An Introduction to Computational Geometry*, M. Minsky and S. Papert Eds., Cambridge, MA, USA: MIT Press, 1969.

[42] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, “Deep learning for computer vision: A brief review,” *Comput. Intell. Neurosci.*, vol. 2018, pp. 1–13, 2018.

[43] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi Eds. Cham, Switzerland:Springer, pp. 234–241.

[44] M. H. Kang, K. Ryu, A. Upadhyaya, and A. Rohatgi, “Optimization of sin ar coating for si solar cells and modules through quantitative assessment of optical and efficiency loss mechanism,” *Prog. Photovolt.: Res. Appl.*, vol. 19, no. 8, pp. 983–990, 2011.

[45] J. B. Schneider, *Understanding the Finite-Difference Time-Domain Method*, School Elect. Eng. and Comput. Sci. Washington State University, Pullman, WA, USA, vol. 28, 2010.

[46] R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer, “A stochastic quasi-newton method for large-scale optimization,” *SIAM J. Optim.*, vol. 26, no. 2, pp. 1008–1031, 2016.

[47] J. Mockus, “Global optimization and the Bayesian approach,” in *Mathematics and Its Applications*. Berlin, Germany: Springer, 1989, pp. 1–3.

[48] F. Caracciolo, E. Dallago, D. G. Finarelli, A. Liberale, and P. Merhej, “Single-variable optimization method for evaluating solar cell and solar module parameters,” *IEEE J. Photovolt.*, vol. 2, no. 2, pp. 173–180, Apr. 2012.

[49] S. Xu, X. An, X. Qiao, L. Zhu, and L. Li, “Multi-output least-squares support vector regression machines,” *Pattern Recognit. Lett.*, vol. 34, no. 9, pp. 1078–1084, 2013.

[50] C. Williams, S. Klanke, S. Vijayakumar, and K. M. Chai, “Multi-task Gaussian process learning of robot inverse dynamics,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 265–272.

[51] E. V. Bonilla, K. M. Chai, and C. Williams, “Multi-task Gaussian process prediction,” in *Advances in Neural Information Processing Systems*, vol. 20. J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds., Red Hook, NY, USA:Curran Associates, Inc., 2008, pp. 153–160.

[52] M. A. Álvarez and N. D. Lawrence, “Computationally efficient convolved multiple output Gaussian processes,” *J. Mach. Learn. Res.*, vol. 12, pp. 1459–1500, 2011.

[53] Ó. García-Hinde, M. Martinez-Ramon, and V. Gómez-Verdejo, “A conditional one-output likelihood formulation for multitask Gaussian processes,” *Neurocomputing*, vol. 509, pp. 257–270, 2022. [Online]. Available: http://dx.doi.org/10.2139/ssrn.4093209

[54] W. Zhang, X. Liu, Y. Ding, and D. Shi, “Multi-output LS-SVR machine in extended feature space,” in *Proc. IEEE Int. Conf. Comput. Intell. Meas. Syst. Appl.*, 2012, pp. 130–134.

[55] G. De’ath, “Multivariate regression trees: A new technique for modeling species–environment relationships,” *Ecology*, vol. 83, no. 4, pp. 1105–1117, 2002.

[56] D. KocevC. Vens, J. Struyf, and S. Džeroski, “Ensembles of multi-objective decision trees,” in *Proc. Eur. Conf. Mach. Learn.*, 2007, pp. 624–631.

[57] A. de Myttenaere, B. Golden, B. Le Grand, and F. Rossi, “Mean absolute percentage error for regression models,” *Neurocomputing*, vol. 192, pp. 38–48, 2016.

[58] M. Belkin, D. Hsu, S. Ma, and S. Mandal, “Reconciling modern machine-learning practice and the classical bias-variance trade-off,” *Proc. Nat. Acad. Sci.*, vol. 116, no. 32, pp. 15849–15854, 2019.

[59] F. D. Foresee and M. Hagan, “Gauss-Newton approximation to Bayesian learning,” in *Proc. Int. Conf. Neural Netw.*, 1997, vol. 3, pp. 1930–1935.

[60] H. B. Demuth, M. H. Beale, O. De Jess, and M. T. Hagan, *Neural Network Design*, 2nd ed. Stillwater, OK, USA: Martin Hagan, 2014.

[61] M. Zagrouba, A. Sellami, M. Bouaïcha, and M. Ksouri, “Identification of PV solar cells and modules parameters using the genetic algorithms: Application to maximum power extraction,” *Sol. Energy*, vol. 84, no. 5, pp. 860–866, 2010.

[62] A. Khare and S. Rangnekar, “A review of particle swarm optimization and its applications in solar photovoltaic system,” *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2997–3006, 2013.

[63] D. M. Kline and V. L. Berardi, “Revisiting squared-error and cross-entropy functions for training neural network classifiers,” *Neural Comput. Appl.*, vol. 14, no. 4, pp. 310–318, 2005.

[64] M. Feurer and F. Hutter, *Hyperparameter Optimization*. Cham, Switzerland:Springer, 2019, pp. 3–33.

[65] I. Fister, I. Fister, X.-S. Yang, and J. Brest, “A comprehensive review of firefly algorithms,” *Swarm Evol. Comput.*, vol. 13, pp. 34–46, 2013.

[66] D. E. Goldberg, “Genetic and evolutionary algorithms come of age,” *Commun. ACM*, vol. 37, no. 3, pp. 113–119, 1994.

[67] M. Li et al., “Numerical simulation of doping process by BBr3 tube diffusion for industrial n -type silicon wafer solar cells,” *IEEE J. Photovolt.*, vol. 7, no. 3, pp. 755–762, May 2017.

[68] H. Li et al., “POCL3 diffusion for industrial Si solar cell emitter formation,” *Front. Energy*, vol. 11, no. 1, pp. 42–51, 2016.

[69] H. Wagner-Mohnsen et al., “Combining numerical simulations, machine learning and genetic algorithms for optimizing a POCL3 diffusion process,” in *Proc. IEEE 48th Photovolt. Specialists Conf.*, 2021, pp. 0528–0531.

[70] Y.-C. Wu and Y.-R. Jhan, *Introduction of Synopsys Sentaurus TCAD Simulation*. Cham, Singapore:Springer, 2018, pp. 1–17.

[71] A. Wilson and R. Adams, “Gaussian process kernels for pattern discovery and extrapolation,” in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 1067–1075.

[72] A. V. Eremeev, “Modeling and analysis of genetic algorithm with tournament selection,” in *Artificial Evolution*, C. Fonlupt, J.-K. Hao, E. Lutton, M. Schoenauer, and E. Ronald Eds. Berlin, Germany:Springer, 2000, pp. 84–95.

[73] J. Murphy, R. McGuire, K. Bothe, V. Voronkov, and R. Falster, “Minority carrier lifetime in silicon photovoltaics: The effect of oxygen precipitation,” *Sol. Energy Mater. Sol. Cells*, vol. 120, pp. 402–411, 2014.

[74] K. Kutsukake, Y. Nagai, T. Horikawa, and H. Banba, “Real-time prediction of interstitial oxygen concentration in Czochralski silicon using machine learning,” *Appl. Phys. Exp.*, vol. 13, no. 12, 2020, Art. no. 125502.

[75] M. H. Sazlı, “A brief review of feed-forward neural networks,” *Commun. Fac. Sci. Univ. Ankara Ser. A2-A3 Phys. Sci. Eng.*, vol. 50, no. 1, pp. 11–17, 2006.

[76] S. Miyagawa, K. Gotoh, K. Kutsukake, Y. Kurokawa, and N. Usami, “Application of Bayesian optimization for improved passivation performance in TIO X/SIO Y/C-Si heterostructure by hydrogen plasma treatment,” *Appl. Phys. Exp.*, vol. 14, no. 2, 2021, Art. no. 025503.

[77] S. Ozturk, E. Kayabasi, E. Celik, and H. Kurt, “Determination of lapping parameters for silicon wafer using an artificial neural network,” *J. Mater. Sci.: Mater. Electron.*, vol. 29, no. 1, pp. 260–270, 2017.

[78] Y. Buratti, C. Eijkens, and Z. Hameiri, “Optimization of solar cell production lines using neural networks and genetic algorithms,” *ACS Appl. Energy Mater.*, vol. 3, no. 11, pp. 10317–10322, 2020.

[79] P. P. Altermatt et al., “A method for optimizing PERC cells in industrial production lines using final IV parameters, statistical procedures and numerical device modeling,” in *Proc. AIP Conf. Proc.*, 2018, vol. 1999, Art. no. 110001.

[80] S. Wasmer, K. Hübener, and B. Klöter, “Explaining the efficiencies of mass-produced p-type CZ-Si solar cells by interpretable machine learning,” *Solar RRL*, vol. 6, no. 5, 2021, Art. no. 2100477.

[81] S. Wasmer and J. Greulich, “Modelling and analysis of solar cell efficiency distributions,” *Jpn. J. Appl. Phys.*, vol. 56, no. 8S2, Jun. 2017, Art. no. 08MB02.

[82] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[83] H. Schwenk and Y. Bengio, “Training methods for adaptive boosting of neural networks,” in *Advances in Neural Information Processing Systems*, M. Jordan, M. Kearns, and S. Solla, Eds., vol. 10. Cambridge, MA, USA:MIT Press, 1997. [Online]. Available: https://proceedings.neurips.cc/paper/1997/file/9cb67ffb59554ab1dabb65bcb370ddd9-Paper.pdf

[84] P. Refaeilzadeh, L. Tang, and H. Liu, *Cross-Validation*. New York, NY, USA:Springer2016, pp. 1–7.

[85] T. Chai and R. R. Draxler, “Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature,” *Geosci. Model Develop.*, vol. 7, no. 3, pp. 1247–1250, 2014.

[86] S. Desale, A. Rasool, S. Andhale, and P. Rane, “Heuristic and metaheuristic algorithms and their relevance to the real world: A survey,” *Int. J. Comput. Eng. Res. Trends*, vol. 351, no. 5, pp. 2349–7084, 2015.

[87] O. Sagi and L. Rokach, “Ensemble learning: A survey,” *WIREs Data Mining Knowl. Discov.*, vol. 8, no. 4, 2018, Art. no. e1249.

[88] A. Çayir, I. Yenidoğan, and H. Dağ, “Feature extraction based on deep learning for some traditional machine learning methods,” in *Proc. 3rd Int. Conf. Comput. Sci. Eng.*, 2018, pp. 494–497.

[89] A. Maćkiewicz and W. Ratajczak, “Principal components analysis (PCA),” *Comput. Geosci.*, vol. 19, no. 3, pp. 303–342, 1993.

[90] F. Murtagh and P. Contreras, “Algorithms for hierarchical clustering: An overview,” *WIREs Data Mining Knowl. Discov.*, vol. 2, no. 1, pp. 86–97, 2012.

[91] W. K. V. Chan, *Theory and Applications of Monte Carlo Simulations*. Rijeka, Croatia: Books on Demand, 2013.

[92] B. Michl et al., “Imaging techniques for quantitative silicon material and solar cell analysis,” *IEEE J. Photovolt.*, vol. 4, no. 6, pp. 1502–1510, Nov. 2014.

[93] M. W. Akram et al., “CNN based automatic detection of photovoltaic cell defects in electroluminescence images,” *Energy*, vol. 189, 2019, Art. no. 116319.

[94] N. Zhang, S. Shan, H. Wei, and K. Zhang, “Micro-cracks detection of polycrystalline solar cells with transfer learning,” *J. Phys.: Conf. Ser.*, vol. 1651, no. 1, Nov. 2020, Art. no. 012118.

[95] S. Deitsch et al., “Automatic classification of defective photovoltaic module cells in electroluminescence images,” *Sol. Energy*, vol. 185, pp. 455–468, 2019.

[96] H. Han et al., “Polycrystalline silicon wafer defect segmentation based on deep convolutional neural networks,” *Pattern Recognit. Lett.*, vol. 130, pp. 234–241, 2020.

[97] P. Kunze, S. Rein, M. Hemsendorf, K. Ramspeck, and M. Demant, “Learning an empirical digital twin from measurement images for a comprehensive quality inspection of solar cells,” *Sol. RRL*, vol. 6, no. 5, 2022, Art. no. 2100483.

[98] A. K. Acharya, P. K. Sahu, and S. R. Jena, “Deep neural network based approach for detection of defective solar cell,” *Mater. Today: Proc.*, vol. 39, pp. 2009–2014, 2021.

[99] B. Su et al., “Deep learning-based solar-cell manufacturing defect detection with complementary attention network,” *IEEE Trans. Ind. Informat.*, vol. 17, no. 6, pp. 4084–4095, Jun. 2021.

[100] M. Demant, P. Virtue, A. Kovvali, S. Yu, and S. Rein, “Deep learning approach to inline quality rating and mapping of multi-crystalline si-wafers,” in *Proc. 35th Eur. Photovolt. Sol. Energy Conf. Exhib.*, 2018, Art. no. 814–818.

[101] M. Demant, P. Virtue, A. Kovvali, S. X. Yu, and S. Rein, “Learning quality rating of as-cut mc-Si wafers via convolutional regression networks,” *IEEE J. Photovolt.*, vol. 9, no. 4, pp. 1064–1072, Jul. 2019.

[102] Y. Buratti, A. Sowmya, R. Evans, T. Trupke, and Z. Hameiri, “End-of-line binning of full and half-cut cells using deep learning on electroluminescence images,” in *Proc. IEEE 47th Photovolt. Specialists Conf.*, 2020, pp. 0133–0138.

[103] Y. Buratti, J. Dick, Q. L. Gia, and Z. Hameiri, “A machine learning approach to defect parameters extraction: Using random forests to inverse the shockley-read-hall equation,” in *Proc. IEEE 46th Photovolt. Specialists Conf.*, 2019, pp. 3070–3073.

[104] Y. LeCun et al., “Backpropagation applied to handwritten zip code recognition,” *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.

[105] Y. LeCun et al., “Generalization and network design strategies,” *Connectionism Perspective*, vol. 19, no. 143–155, 1989, Art. no. 18.

[106] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. 3rd Int. Conf. Learn. Representations*, 2015, pp. 143–155.

[107] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Trans. Knowl. data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2009.

[108] M. Long, Y. Cao, J. Wang, and M. I. Jordan, “Learning transferable features with deep adaptation networks,” in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn.*, 2015, vol. 37, pp. 97–105.

[109] X. Zhu, K.-H. Thung, E. Adeli, Y. Zhang, and D. Shen, “Maximum mean discrepancy based multiple kernel learning for incomplete multimodality neuroimaging data,” in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2017, pp. 72–80.

[110] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *Adv. Neural Inf. Process. Syst.*, vol. 28, 2015.

[111] G. Huang, Z. Liu, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.

[112] D. Chicco, “Siamese neural networks: An overview,” *Artificial Neural Networks*. Berlin, Germany:Springer, 2021, pp. 73–94.

[113] A. Vaswani et al., “Attention is all you need,” in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2017, vol. 2190, pp. 73–94.

[114] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2921–2929.

[115] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *J. Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[116] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, vol. 25, pp. 1–48.

[117] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[118] J. L. Speiser, M. E. Miller, J. Tooze, and E. Ip, “A comparison of random forest variable selection methods for classification prediction modeling,” *Expert Syst. Appl.*, vol. 134, pp. 93–101, 2019.

[119] J. Friedman, T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu, “Discussion of boosting papers,” *Annu. Statist.*, vol. 32, pp. 102–107, 2004.