Energy Conversion and Management 291 (2023) 117263

Contents lists available at ScienceDirect

Energy Conversion and Management

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

Artificial intelligence for solid oxide fuel cells: Combining automated high accuracy artificial neural network model generation and genetic algorithm for time-efficient performance prediction and optimization

Felix Mütter [*], Clemens Berger, Benjamin Königshofer, Michael Höber, Christoph Hochenauer, Vanja Subotić

Institute of Thermal Engineering, Graz University of Technology, Inffeldgasse 25/B, 8010 Graz, Austria

ARTICLE INFO

Keywords:
Artificial neural network
Solid oxide fuel cell
Optimization
Time-efficient prediction
Genetic algorithm

ABSTRACT

In order to accelerate the commercialization of solid oxide fuel cells, optimal process parameters for reliable and efficient electricity generation are of the highest interest. To reduce the number of timely and monetarily expensive experiments used to find suitable operating parameters, while also allowing for only small sacrifices in accuracy, an artificial neural network (ANN) is used in combination with algorithm-based optimization in this study. An ANN was trained with data from a complex multi-physics model and coupled with a genetic algorithm (GA) to find the maximum power output within a range of operational parameters. Instead of time-consuming, manual fine-tuning of the ANN's model architecture and hyperparameters (HP), a Bayesian HP-tuning algorithm is used in this work. To avoid over-fitting and to ensure a high model consistency, nested k-fold cross validation is implemented. Very low error values of the ANN are achieved both in the k-fold cross-validation and in an additionally performed validation by means of experimental data and a computational fluid dynamics simulation. Compared to a multi-physics model used to generate training data, the ANN achieved an increased prediction speed of more than three orders of magnitude, while only minimal decreases to prediction accuracy. Optimization with the GA produced consistent results close to the global optimum and also provided good alternative solutions with significantly different gas compositions at high power. The validity of the solutions found with the GA was underpinned with the help of a sensitivity analysis, which was carried out for the most promising SOFC operating case.

1. Introduction

The transition to carbon-free power generation technologies is an enormously important step towards reducing greenhouse gas emissions in the face of climate change. With their high efficiency and fuel flexibility, solid oxide fuel cells (SOFCs) are a viable technology in this process. Compared to conventional technologies like internal combustion engines and gas turbines, solid oxide fuel cells are not limited by the Carnot efficiency and can be operated with a variety of fuels, ranging from conventional natural gas over biogas and ammonia to hydrogen. Furthermore, due to its high operation temperature of 600–900 °C, precious metals are not required as in proton exchange membrane fuel cells [1,2].

Potential areas of application range from small scale SOFC systems as auxiliary power units to applications for decentralized heat and electricity supply for single-households or large scale power plants in combination with gas turbines [3,4].

To utilize the full potential of this technology and ensure its stable, safe and efficient long-term operation, further research must be conducted on the successful commercialization of this technology. Due to the large number of potential operation parameters, experimental SOFC performance evaluations are very time-consuming and expensive. To reduce both the timely and monetarily efforts that are required to figure out optimal operation parameters, modelling is a valid approach [5]. To predict the performance of an SOFC, modelling approaches with different levels of complexity, ranging from 0D to 3D, exist. With the lowest complexity 0D-models do not consider the cell size and the spatial distribution of current densities and local species concentrations. When taking into account models that predict voltage in the literature, a voltage deviation of 5% [6] and as low as 2.47% [7] compared to experimental data can be found for those models. 1D SOFC-models provide detailed information regarding porous media transport, mass

[*] Corresponding author.
E-mail address: felix.muetter@tugraz.at (F. Mütter).

https://doi.org/10.1016/j.enconman.2023.117263
Received 10 March 2023; Received in revised form 2 June 2023; Accepted 3 June 2023
Available online 17 June 2023

transport and heterogeneous chemistry, whereby in the literature voltage deviations of 3 - 2.38% compared to experimental data have been published [8,9]. To gain a more precise insight in the behaviour of SOFC 2D models, which allow for the simulation of spatial changes in temperature, fuel and air side species concentrations, as well as current density changes, can be used. Thereby, relative errors in the range of 1.5%–4% can be found in the literature [10–12]. With 3D-CFD models, the prediction error can be further decreased and voltage deviations of as low as 0.5% compared to experiments have been achieved [13].

The higher accuracy of high dimensional models (2D and 3D) is offset by their higher computational costs [14]. Ideally would be a model with the accuracy of an experiment and a required computation time in the range of milliseconds. With such a model, the usage of computational expensive optimization algorithms, as well as real-time predictions for model-predictive control, would be feasible. Conventional model-based simulation approaches suffer from the aforementioned accuracy/computational cost trade-off. This is where machine learning (ML) comes into play. In contrast to the model-based approach of physical models, data-driven ML approaches, like ANN, do not require a detailed knowledge of the underlying processes and thus require less complex calculations for its predictions. This reduced complexity results in a significant reduction of computation time. With proper model training and high-quality training data, the achievable prediction accuracies might come out only slightly lower than that of the training data, thus closing the gap between high-precision and high-speed predictions. Furthermore, the source of the data is not just limited to simulated data. This allows models to be developed from both experimental and synthetic data. Models can even use transfer learning and start training with synthetic data, thus fine-tuning the model by means of experimental data [15].

ANNs have already been used to successfully predict high temperature solid oxide cells in both fuel cell and electrolysis mode, e.g., the authors in [16] developed a two-layer feed-forward network to predict the outlet gas composition, current density and temperatures. To optimize the ANN network architecture, the learning rate, number of hidden neurons and number of training epochs were altered in a trial-and-error approach. The evaluation of the prediction performance was facilitated by means of a random test data set. In [17], several different ANN model architectures were trained and evaluated by means of experimental training data and a random data split validation. Instead of tuning the model architecture, the performance of 10 different architectures was evaluated and compared, whereby models with a higher number of hidden layers (≥3) exhibited a better prediction performance than shallower (<3 hidden layers) architectures. The authors in [18] successfully modelled a solid oxide electrolyzer using an extreme learning machine. Here, only the number of hidden nodes for the hidden layer had to be specified by the user before model training. Not just the cell voltage but also the prediction of the cell impedance from electrochemical impedance spectroscopy measurements of an SOFC was successfully achieved with the aid of neural networks [19]. The application of data-driven surrogate models is not limited to solid oxide cells. Deng et al. predicted and optimized the gas distribution quality of a high-temperature proton exchange membrane fuel cell through a deep belief network optimized by a genetic algorithm [20]. The authors in [21] successfully predicted the energy consumption of a range extender fuel cell hybrid vehicle, and Wang et al. used several ML-techniques to optimize the energy management of a fuel cell vehicle [22].

Due to the significant decrease in prediction time (≪1 s/prediction), the usage of otherwise overly time-consuming optimization algorithms, like genetic algorithms, is possible with ANN. In the literature, already several contributions regarding ANN in combination with optimization algorithms can be found. A combination of artificial neural networks and genetic algorithms has been used successfully for the optimization of SOFC design parameters like layer thicknesses, porosities [23,24] or manufacturing parameters like sintering temperature [25]. The authors in [26] successfully coupled a neural network, which was trained with data from a multi-physics model, with a genetic algorithm for SOFC performance optimization.

What all the above listed sources share is the need for proper hyperparameter tuning, which is usually done through time-consuming, manual fine-tuning. This work successfully automatized this process by means of optimization algorithms. A special focus was set on the proper validation of a trained ANN to ensure good generalization and model consistency. With the commonly used random split validation, no information regarding consistency is provided. Therefore, a repeated nested k-fold cross-validation was used by the authors as suggested by [27]. Despite the additional computational effort induced by introducing repeated nested k-fold cross-validation, the extra effort immediately paid off by revealing insufficient model consistency that would not have been detected without this approach.

Stacking was introduced to increase model consistence. This measure led to a significant increase in both model prediction performance and consistency. To the authors best knowledge, the modelling approach demonstrated in this publication using nested k-fold cross-validation, automated HP-optimization, and stacking have not been previously published in scientific papers in the context of SOFC modelling of industrial-scale anode-supported cells.

After the successful training of the ANN model, the prediction performance was compared with experimental data and a 2D-CFD simulation with both showing very good agreement. The trained ANN model was then successfully coupled with a genetic algorithm to maximize the SOFC power output. The optimization led to consistent and plausible results close to the global optimum, and furthermore, offered alternative process parameters with performance close to the global optimum but significant different gas compositions. To qualitatively validate the results obtained with the genetic algorithm, a sensitivity analysis was performed for the most promising operating case. For a better overview, all process steps that were carried out in the course of this work are summarized as a flow chart in Fig. 1.

2. Data generation

The data used for training the ANN model was generated by means of a 2D-multi-physics model. This model is based on a 2D multi-physic SOFC model with more than 120 equations, which is capable of predicting the cell voltage for non-carbonaceous species [12]. This model was further enhanced to incorporate internal reforming of CH_4, the electrochemical and chemical reactions as well as the loss mechanisms that take place when dealing with carbon species in an SOFC. For this work, an industrial sized anode-supported cell (ASC) with an active surface area of 80 cm^2 and a 300 μm nickel/yttria-stabilized zirconia (Ni/YSZ) anode was simulated. This cell type was chosen because the data from in-house experiments and results of a 2D-CFD simulation were readily accessible for model validation.

2.1. Model configuration

In Fig. 2, the general principle of the 2D-multi-physics model for the data generation is visualized. In step one (“Plug Flow Reactor Step 1” in Fig. 2), the local chemical equilibrium in the fuel flow channel for a given fuel composition, temperature, pressure and fuel mass flow rate is computed with the aid of the multi-step surface reaction mechanism for steam reforming of methane over nickel, as described in [28]. This step reflects potential ongoing internal reforming. After this, based on the local chemical equilibrium and further input parameters (e.g., current and temperature), the following values are calculated (“Electrochemical Model (Dusty Gas) Step 1” in Fig. 2):

- Species production rates
- Composition dependent diffusion parameters
- Nernst voltage, ohmic-, activation-, and concentration losses

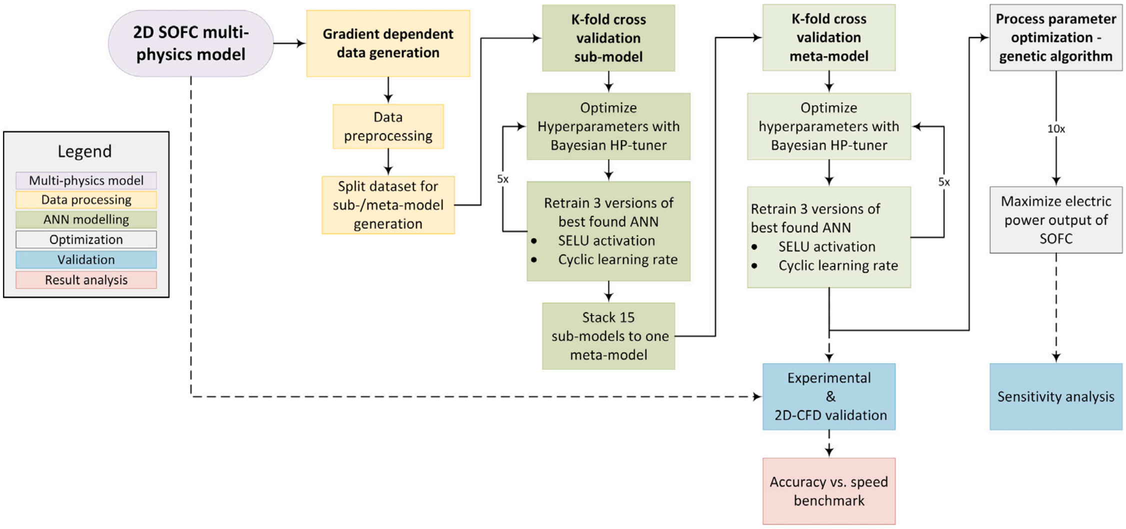
2D SOFC multi-physics model
Gradient dependent data generation
Data preprocessing
Split dataset for sub-/meta-model generation
K-fold cross validation sub-model
Optimize Hyperparameters with Bayesian HP-tuner
5x
Retrain 3 versions of best found ANN
SELU activation
Cyclic learning rate
Stack 15 sub-models to one meta-model
K-fold cross validation meta-model
Optimize hyperparameters with Bayesian HP-tuner
5x
Retrain 3 versions of best found ANN
SELU activation
Cyclic learning rate
Process parameter optimization - genetic algorithm
10x
Maximize electric power output of SOFC
Experimental & 2D-CFD validation
Sensitivity analysis
Accuracy vs. speed benchmark
Legend
Multi-physics model
Data processing
ANN modelling
Optimization
Validation
Result analysis

Fig. 1. Flow chart showing the basic principle of ANN modelling and process parameter optimization.

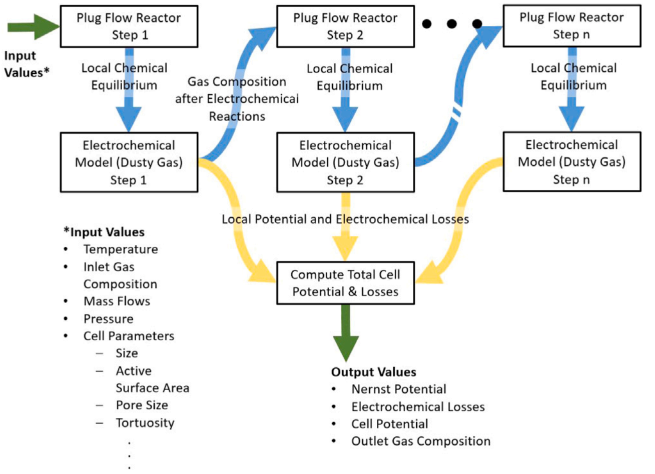
Input Values*
Plug Flow Reactor Step 1
Local Chemical Equilibrium
Electrochemical Model (Dusty Gas) Step 1
Gas Composition after Electrochemical Reactions
Plug Flow Reactor Step 2
Local Chemical Equilibrium
Electrochemical Model (Dusty Gas) Step 2
Plug Flow Reactor Step n
Local Chemical Equilibrium
Electrochemical Model (Dusty Gas) Step n
Local Potential and Electrochemical Losses
Compute Total Cell Potential & Losses
*Input Values
Temperature
Inlet Gas Composition
Mass Flows
Pressure
Cell Parameters
Size
Active Surface Area
Pore Size
Tortuosity
Output Values
Nernst Potential
Electrochemical Losses
Cell Potential
Outlet Gas Composition

Fig. 2. General principle of 2D-PFR-multi-physics model. The local chemical equilibrium is computed from the input values in the first step. These values are then used to solve the electrochemical equations. The gas composition is then updated to reflect its state after the electrochemical reactions occurs and is fed into the next plug flow reactor. The overall cell potential and losses are calculated from the local values in the final step.

Based on these calculations, the local gas composition in both the fuel and air flow channels is updated to the state after all local reactions have reached steady state and fed into the next reactor ("Plug Flow Reactor Step 2" in Fig. 2). This procedure is repeated until all local cells have been computed. In a last step, from all the local calculations, the overall cell voltage and cell losses are calculated. Fig. 3 shows the flow configuration of the 2D multi-physics model. For the simulations, the flow channels for the fuel side and the air side are represented by 100 plug flow reactors lined up per flow channel.

2.2. Operation parameter range

Broad ranges for the training dataset parameters were defined to ensure that all the operation modes relevant to the SOFC are included, as the extrapolation capabilities of standard (i.e., physic-less) neural networks are limited [29]. The resolution of the dataset was defined in a way so that non-linear effects of higher order are also represented by the dataset. To achieve this, the minimum number of steps per parameter was set to 5 for the temperature and the species CO, CO_2 and CH_4. To keep the resolution in the same range for H_2 and H_2O, 10 steps within the defined range were computed. N_2 was used as balancing species in case the sum of all other species did not add up to 100%.

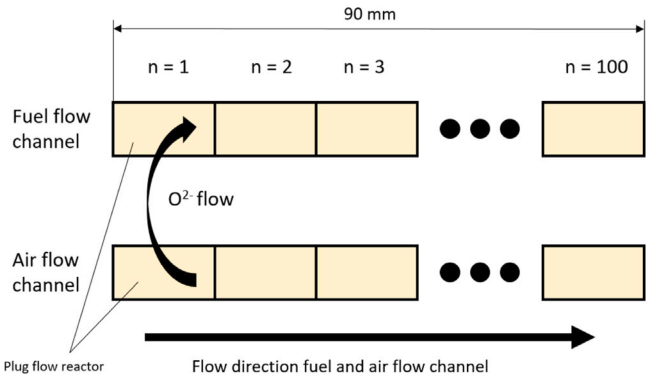

Fig. 3. Configuration of the 2D multi-physics model.

Table 1
Molar fraction, temperature and current density range of dataset generated with the 2D multi physics model for ANN training.

	Range	Interval	Steps	Unit
H_2	5–95	10	10	mol-%
H_2O	5–95	10	10	mol-%
CO	0–40	10	5	mol-%
CO_2	0–40	10	5	mol-%
CH_4	0–10	2	6	mol-%
N_2	Balance	–	–	mol-%
Temperature	700–800	25	5	°C
Current density	0–617.3	Gradient dependent	52	$\frac{mA}{cm^2}$

2.2.1. Gradient dependent resolution refinement

A gradient dependent interval length was implemented for the current density. This measure was taken to increase the resolution of the dataset in the non-linear region of the polarization curve at higher current densities. Because the area of non-linearity is strongly dependent on fuel composition and thus not always at the same current density values, the following steps were taken during the generation of datapoints:

First, the voltage is computed at a current density of 617.3 $\frac{mA}{cm^2}$. If the voltage is less than 0.6 V, the maximum allowable current density is sought by successive halving to achieve 0.6 V. This step was necessary –

especially for fuel compositions with small proportions of combustible gases – because the upper limit of the current density can never be reached due to fuel starvation. The lower voltage limit of 0.6 V was chosen so that the ANN model can also represent the range of >0.7 V where accelerated degradation due to Ni-reoxidation may occur [30,31]. Once the maximum allowable current density is determined, 25 uniformly distributed points are placed within the specified upper and lower current density limits. After this point, all datapoints are evaluated using Eq. (1). From here, at the position with the highest deviation from the linear approximation dev_n two new datapoints at the position $n+1$ and $n-1$ are added.

$$dev_n = |\frac{E_{n-1}+E_{n+1}}{2} - E_n| \tag{1}$$

This procedure is repeated until a total number of 52 datapoints for one gas composition and temperature is reached.

In Fig. 4, a polarization curve was generated from this approach. With the ranges and number of steps defined in Table 1, the overall number of datapoints concluded to 534 976 individual entries, which is equivalent to 10 288 unique polarization curves. For the generation of the ANN training dataset, all together, 22 CPU cores from three different computers were used, requiring a rough total of 270 h to generate the whole dataset.

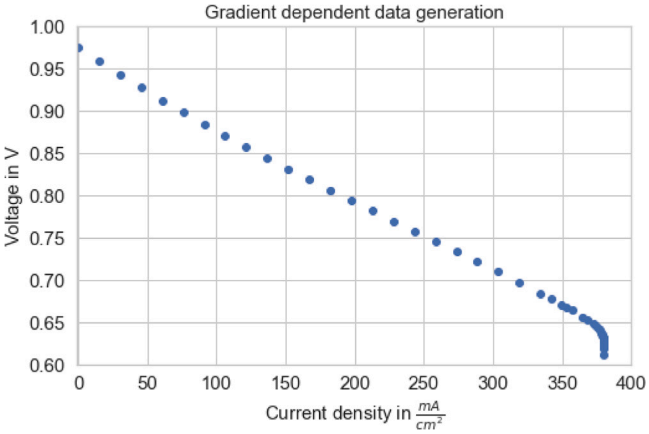

Fig. 4. Polarization curve generated from the described approach. The point density in the non-linear region of the curve's right side is higher than the linear section.

2.3. Data preprocessing

For model training, several preprocessing steps for the created dataset were taken, namely:

- Filtering
- Shuffling
- Normalization

Filtering excluded all dataset entries with a voltage below 0 V or above 1.229 V as those values above the oxygen evolution reaction potential would be implausible for SOFC operation mode [32]. Shuffling was implemented to reduce the risk of a biased model due to the sequence of training data fed to the ANN during training. Normalization was introduced to reduce the risk of unstable model training due to too high model weights caused by too excessive input feature values. Therefore, all input feature values were rescaled to be within the range of 0 to 1.

3. Artificial neural network modelling

3.1. Implemented methods and functions

The neural network model was implemented in Python3 by means of Tensorflow and its API Keras [33,34]. Algorithm-based hyperparameter (HP) tuning was implemented to increase the model prediction performance [35]. To ensure the most unbiased assessment of model performance, a nested k-fold cross validation was carried out by means of scikit-learn [36]. With this approach, not just the model performance, but also the standard deviation of the model performance could be evaluated. To further increase the model prediction performance and the model consistency stacking was implemented. For a faster and better learning procedure, a cyclic learning rate to adopt the learning rate throughout the model training, was used [37]. Several different activation functions were investigated, whereby "scaled exponential linear units" (SELUs) yielded the best results [38]. The following section describes in detail all the previously mentioned functions that were used to set up the ANN model and the training procedure.

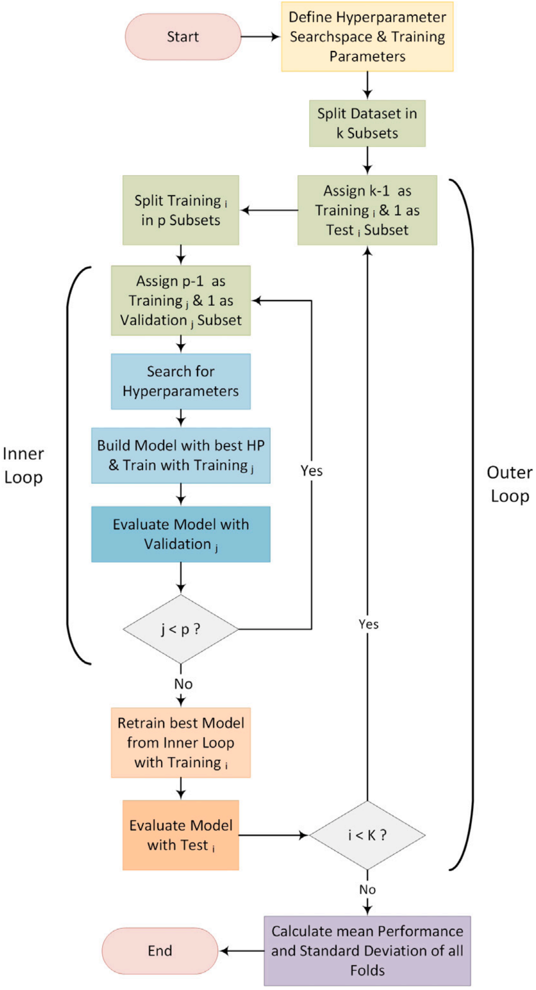
Start
Define Hyperparameter Searchspace & Training Parameters
Split Dataset in k Subsets
Assign k-1 as Training $_i$ & 1 as Test $_i$ Subset
Split Training $_i$ in p Subsets
Assign p-1 as Training $_j$ & 1 as Validation $_j$ Subset
Search for Hyperparameters
Build Model with best HP & Train with Training $_j$
Evaluate Model with Validation $_j$
j < p ?
Yes
No
Inner Loop
Retrain best Model from Inner Loop with Training $_i$
Evaluate Model with Test $_i$
i < K ?
Yes
No
Outer Loop
Calculate mean Performance and Standard Deviation of all Folds
End

Fig. 5. Simplified flow chart for training, validating and testing of the ANNs with nested k-fold cross validation.

3.1.1. Nested k-fold cross validation

The whole modelling and validation process consists of an outer and an inner loop. The task of the outer loop is to evaluate the model performance, while the inner loop is used to optimize the HPs of the ANN. With this set up, the data leakage that would otherwise occur can be avoided. After an initial split of the data into k subsets, k–1 are handled as training subsets and the remaining subset is set as test dataset in the outer loop. The training set is fed to the inner loop, where it is again split into $p-1$ training subsets and 1 validation subset. With this data configuration, promising HPs are searched for by means of an optimization algorithm. With the result from the HP search, an ANN-model is trained and evaluated with the corresponding validation dataset. Finding optimal HPs is repeated for every possible configuration of training and validation datasets and thus repeated p times for each iteration of the outer loop. The best performing configuration from those p results is selected and retrained with the training set of the outer loop and evaluated with the set aside test set. The k iterations of the outer loop led to k different models and thus k different performance values. The average model performance and the standard deviation from all models can be calculated from these results, giving insights in the stability of the whole modelling approach. Fig. 5 shows the flow chart of the implemented nested k-fold cross validation.

3.1.2. Stacking

First investigations with the nested k-fold cross validation approach revealed an undesirably high deviation in terms of model performance. As a countermeasure, stacking was introduced. Stacking is an ensemble method in which several sub-models are combined into a new, larger model that predicts the output value based on the results of each individual sub-model. Fig. 6 shows the schematic principle of a stacked model. A series of individual level-0 models use the same input values to predict the voltage. These predictions are then fed to the level-1 model which makes one final prediction based on all the voltage predictions from the sub-models.

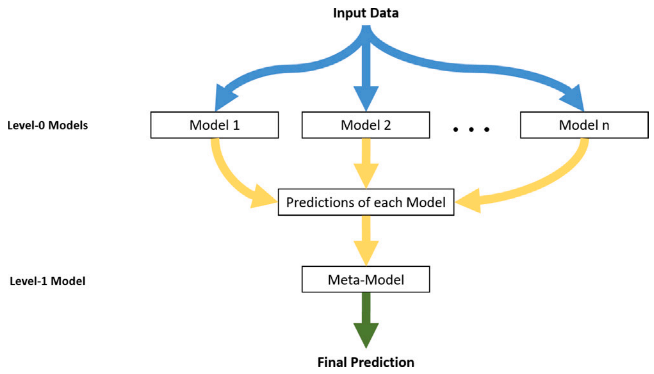

Fig. 6. Stacking schematics.

In the course of this work, k was defined as 5. To gain additional insight into model consistency, 3 versions of the best found model architecture of each fold (see orange boxes in Fig. 5) were trained with the corresponding training dataset. This created a total of 15 sub-models for stacking. The same approach was taken for the stacked models, resulting in 15 meta-models with 5 unique architectures.

3.1.3. HP optimization algorithm

Instead of manually fine-tuning the hyperparameters of the ANN via a trial-and-error approach, an optimization algorithm was implemented to automate this time-consuming step. Two algorithms were considered: HyperBand and Bayesian optimization [39]. Moreover, for the purposes of comparing these two methods directly, a test run with identical parameters was carried out. The results showed that the MSE values of the Bayesian algorithm were better and more consistent, so it was chosen over HyperBand. Through further fine-tuning, an improvement of the MSE value by two orders of magnitude could be achieved. In Fig. 7, the results of the HP-tuner comparison can be seen. Bayesian optimization is a probabilistic optimization strategy for finding the extrema of expensive to evaluate objective functions [40]. It contains two major components, a Gaussian process regressor and an acquisition function. The Gaussian process regressor makes use of the Bayes theorem to approximate the objective function within a range of uncertainty. From this information, the acquisition function is calculated to determine where to place the next sample point in the parameter scan range. See [40–42] for more details on the Bayesian optimization.

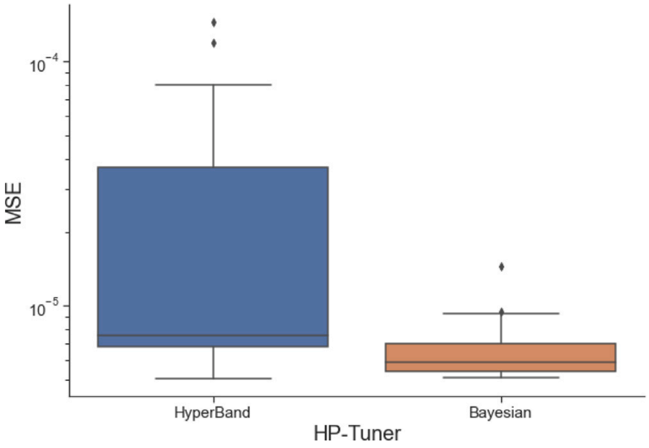

Fig. 7. Results of direct HP-tuner comparison.

3.1.4. Cyclic learning rate

For model training, CLR was introduced instead of the fixed learning rate that was used for the HP-optimization. Cyclic learning rate, suggested by [37], is an approach that combines the advantages of a large learning rate (fast convergence) with the benefits of a small learning rate (better performance) through varying the learning rate between reasonable boundaries throughout the training. As depicted in Fig. 8, a triangular schedule with an exponential decay was chosen for the model training in this study. Introducing an oscillating learning rate scheme, as opposed to a fixed learning rate scheme, led to a further increase in prediction performance and resulted in a reduction of the test-set MSE values of roughly one order of magnitude. Furthermore, due to faster convergence, it reduced the computation time during training.

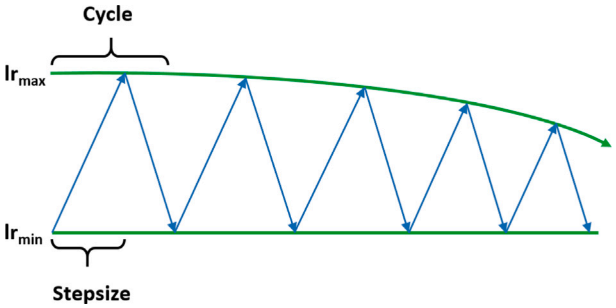

Fig. 8. Learning rate schedule with exponential decay adopted from [37].

3.1.5. Activation function

Six different activation functions were investigated during model set up, namely: sigmoid, tanh, exponential linear unit (Elu), Rectified Linear Unit (ReLU), Swish and SELU [34,38,43]. A visualization of all the investigated activation functions can be seen in Fig. 9.

Initially, ReLU led to the best results, but not consistently. SELU activation was implemented as a substitution for ReLU because it led to a phenomenon called dying-ReLU problem. This causes a deactivation of most of the nodes until it finally leads to a full deactivation of the network [44]. Despite being more computationally expensive, SELU offers the advantage of self-normalization which avoids exploding and vanishing gradients [38].

3.2. Training of artificial neural networks

The purpose of training an ANN is to make this universal framework match a specific case and mimic the behaviour of the underlying

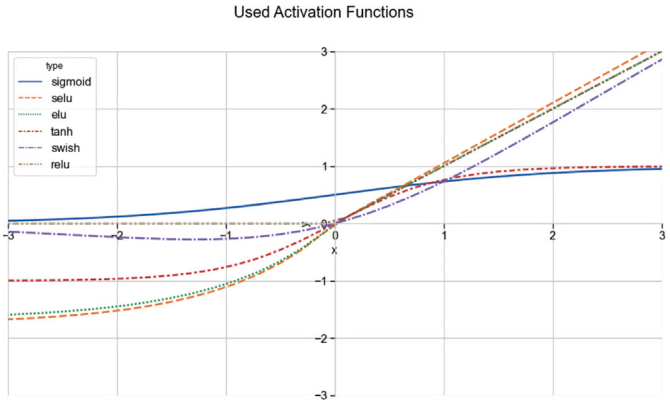

Fig. 9. Visualization of all the investigated activation functions.

system, all with the least possible amount of deviation. In this study, the training of the ANN is carried out by means of the back-propagation algorithm: (i) In a forward step, input values are fed to the ANN to make a prediction y_i based on the weights and biases of each individual network node. (ii) This prediction is then processed in a loss function with the known output value from the training dataset $\hat{y_i}$ to evaluate the quality of the prediction. (iii) In a backward step, the value of each node and bias is adopted depending on its share on the prediction error. This process is repeated until the prediction error meets a predefined threshold value or the maximum number of iterations is reached. The loss function used in this study was mean squared error (MSE) as it penalizes outliers more strictly and thus should lead to a more consistent model:

$$MSE = \frac{1}{n}\sum_{1}^{n}(y_i - \hat{y_i})^2 \tag{2}$$

During model training, it is especially important to terminate training at the right number of iterations. Not enough training leads to underfitting, resulting in too high a training error. When training continues for too long, overfitting may occur. An overfitted model reflects not just the underlying system behaviour but also the inevitable training data noise, which will again lead to poor prediction performance. To avoid overfitting, early stopping was implemented in this study. This measure terminates model training in case the loss of the validation dataset starts to increase. As the errors during model training may vary, patience was introduced as additional HP. Patience describes the maximum number of iterations where the validation error may be higher than the lowest value achieved. With this, local optima during model training may be surpassed and better model performances closer to the global optimum potentially achieved.

3.2.1. ANN model training parameters

The constraints for the HP search and model training are summarized in Tables 2–4.

Table 2
Training parameters.

Parameter	Model 1
Patience	500
Max # of Epochs	10 000
lr_{min}	$1 \cdot 10^{-6}$
lr_{max}	$5 \cdot 10^{-3}$
Batch size	512

In Table 2, the parameters for model training are listed. The borders for the learning rate lr_{min} and lr_{max} were defined through a manual test run as suggested in [37]. The maximum number of epochs was defined with 10 000 to ensure that there were enough epochs for proper convergence. With the defined patience value of 500 epochs, overcoming local minima during training should be enabled. A trade-off between training speed and model performance was made, and the batch size was set to a value of 512 for all model trainings and the HP-optimization.

Table 3
Scanned range and parameters for HP optimization based on Bayesian HP-tuner.

Parameter	Scanned range
Number of nodes per layer sub-model	2–50
Number of layers	2–5
Batch size	512
Weight initialization	accord. [38]
Learning rate	0.005–0.05
Dropout rate	0.0–0.4
Activation function	SELU
Number of nodes per layer meta solver	2–200
HP-Tuner	Bayesian
Number of trials HP-tuner	100
Number of epochs HP-tuner	350

For HP-optimization, a constant learning rate was defined to make results comparable. The range of the learning rate was defined to compensate for slower convergence induced by the dropout layers. These layers act as a regularization measure and should lead to a more robust model and reduce the risk of overfitting through the temporary deactivation of nodes during model training [45].

Table 4
Global and cross validation parameters.

Parameter	Value
Number of folds inner loop	4
Number of folds outer loop	5
Number of models per outer fold	3
Ensemble/Meta-Solver Data Split	70/30

The computations were executed on a 12-core Intel Core i9-7920X CPU, leading to an overall model training time of roughly 60 h.

4. Genetic algorithm implementation

A genetic algorithm (GA) is an optimization algorithm that is based on the process of natural selection and evolution. In a genetic algorithm, like in nature, a certain number of solution configurations form a population. Each individual has a set of features which makes it more or less valuable. In an iterative process, the following steps are repeated until a certain stop criteria is met [46]. The flowchart describing the general principle of a genetic algorithm can be seen in Fig. 10.

(1.) **Selection** The most valuable fraction of the population is selected to form descendants, and the worst performing fraction of the population is discarded.
(2.) **Recombination** The selected individuals from (1.) are merged into parents that combine their features and create new population members of itself until the population limit is reached again.
(3.) **Mutation** Some features of the mated population may randomly mutate to different values.
(4.) **Evaluation** Each individual in the population is evaluated according to a user-defined fitness criterion.

As a search space for the parameter optimization, the same range as was used for the ANN training data was defined. In Table 5, the scan range limits for the GA are summarized. For an evaluation of the individual solution configurations of the GA, the following fitness function was defined to maximize the cell power output:

$$F_n = -E_n \cdot i_n \cdot A + \sum_{n=1}^{n=3} p_n \tag{3}$$

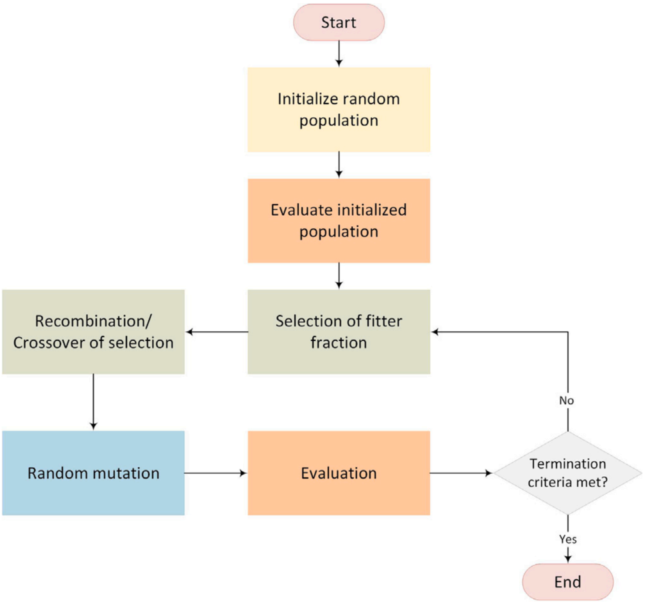
Start
Initialize random population
Evaluate initialized population
Selection of fitter fraction
Recombination/ Crossover of selection
Random mutation
Evaluation
Termination criteria met?
No
Yes
End

Fig. 10. Simplified flow chart of genetic algorithm.

Thus, E_n is the voltage predicted from the ANN, i_n is the current density, A is the active cell area of the SOFC single cell and $\sum_{n=1}^{n=3} p_n$ is the sum of the three penalties that were implemented to punish violations of the introduced boundary conditions. Because the GA implementation in python treats every task to solve as a minimization problem and optimizing the power output is a maximization problem, the voltage value in the fitness function is multiplied by −1 in Eq. (3). Three constraints were defined to ensure a stable convergence of the algorithm:

(1.) **Conservation of masses** - The sum of all fuel fractions must be equal to 1. Because N_2 was used to balance fuel mixtures the requirement for mass conservation was defined as:

$$\begin{aligned} &\text{if} \quad x_{H_2} + x_{H_2O} + x_{CO} + x_{CO_2} + x_{CH_4} > 1 : \\ &p_1 = ((1 - (x_{H_2} + x_{H_2O} + x_{CO} + x_{CO_2} + x_{CH_4})) * 6000)^2 \end{aligned} \tag{4}$$

(2.) **Avoid Ni-oxidation** - If the cell voltage E falls below 0.65 V Ni-oxidation may occur:

$$\begin{aligned} &if \quad E < 0.65 : \\ &p_2 = ((0.65 - E) * 1000)^2 \end{aligned} \tag{5}$$

(3.) **Limit max. fuel utilization** - A too high fuel utilization u_f may lead to undesirable local fuel starvation:

$$\begin{aligned} &if \quad u_f > 0.9 : \\ &p_3 = (u_f \cdot 1000)^2 \end{aligned} \tag{6}$$

The utilization function u_f thereby is defined as:

$$u_f = \frac{i_n \cdot F \cdot A}{n_{fuel} \cdot (2 \cdot x_{H_2} + 2 \cdot x_{CO} + 8 \cdot x_{CH_4})} \tag{7}$$

Whereby F is the Faraday constant and n_{fuel} is the molar fuel flux. In Eqs. (4)–(7) x_n indicates the molar fraction of the respective species n in the fuel feed stream.

The genetic algorithm optimization was executed by means of the genetic algorithm package in Python3 [47]. In Table 6 the used parameters for the optimization are listed. An elitist genetic algorithm was used in this work. With an elitist ratio of 5% and a population size of 20, the best performing member of an iteration is always kept within the population, thus leading to an always non-increasing convergence curve.

Table 5
Scan range limits for genetic algorithm process parameter optimization.

	Min	Max	Unit
H_2	5.0	95.0	mol-%
H_2O	5.0	95.0	mol-%
CO	0.0	40.0	mol-%
CO_2	0.0	40.0	mol-%
CH_4	0.0	10.0	mol-%
N_2	Balance	Balance	mol-%
Temperature	700	800	°C
Current density	0	600	$\frac{mA}{cm^2}$

Table 6
Genetic algorithm parameters.

Parameter	Value	Unit
Max iterations	1500	–
Population size	20	–
Mutation probability	2	%
Elitist ratio	5	%
Crossover probability	50	%
Parent portion	35	%
Crossover type	Uniform	–
Max. iteration w.o. improvement	200	–

5. Results and discussions

In this section, the results obtained employing the ANN-based polarization curve model and the GA-based process parameter optimization are analysed and discussed in detail. At first, the ANN-model's performance is assessed through the nested k-fold cross validation test datasets. Additionally, the ANN-model performance is benchmarked with a CFD analysis and experimental cell tests conducted in-house. A comparison of the required computation time against the model accuracy with values from literature and the results from the genetic algorithm operation parameter optimization conclude this section.

5.1. *Artificial neural network*

The developed ANN-model consists of 8 input parameters, namely the cell temperature, the operating current, the molar fractions of fuel gas species H_2, H_2O, CO, CO_2, CH_4 and N_2, and the voltage as output parameter. The results of the hyperparameter optimization executed in the inner loop of the nested k-fold cross validation approach can be seen in Table 7 — both for the sub-models and the meta-model. Fig. 12 shows a representation of the final model.

Table 7
Results of hyperparameter optimization.

Parameter		Fold				
		1	2	3	4	5
	Input	50	41	50	50	42
	1	30	50	50	50	50
Sub-model	2	47	50	50	50	50
Layers # of nodes	3	50	6	32	50	50
	4	2	2	50	2	5
	5	2	50	2	2	2
Meta model # of nodes		200	84	2	176	196
Learning rate				0.005		
Dropout rate				0.0		

All five folds of the k-fold cross validation for the sub-model and the meta model share a commonality: that the minimum learning rate, together with no dropout, yielded the best performing ANN-model. For the sub-model structure, a tendency to hit both the maximum as well as the minimum number of nodes per layer can be seen. Thus, increasing the scan range for further improvements would be an option.

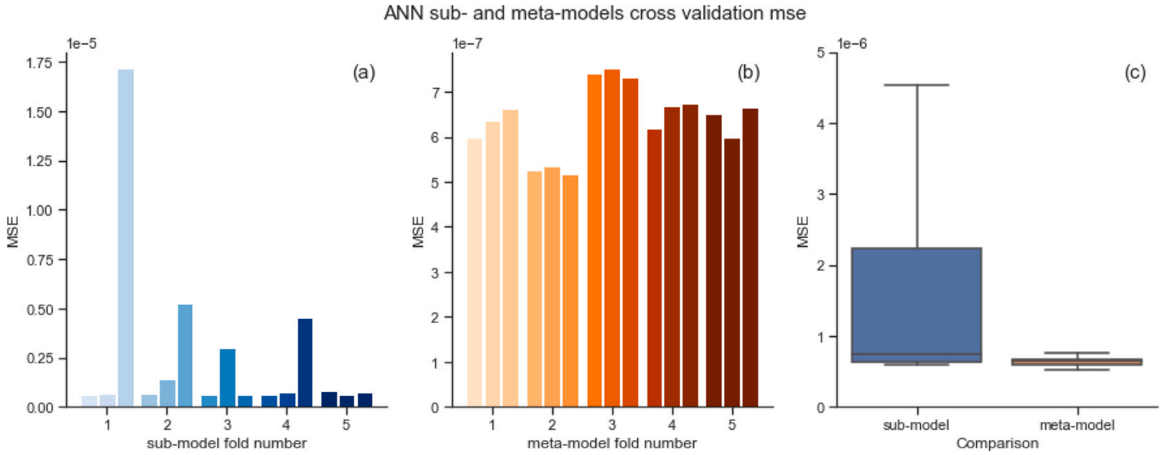

Fig. 11. MSE of (a) ANN sub-models, (b) ANN meta-models from cross validation with test dataset. (c) compares the overall mse boxplot for all the ANN sub-models (left) and all the ANN meta-models (right).

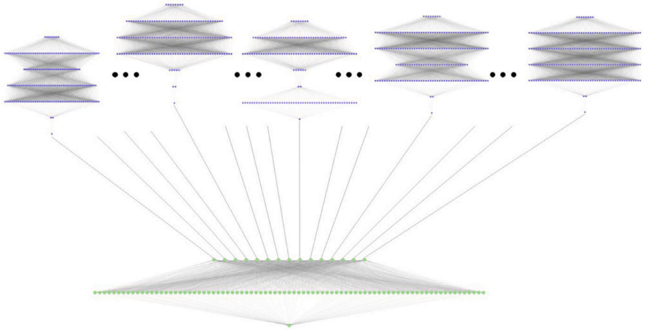

Fig. 12. Model architecture consisting of sub-models (purple nodes) and meta-model (green nodes). Only the unique sub-models are visualized. The sub-models with identical architecture are represented by the full lines pointing upwards.

Table 8
Mean squared error of individual sub-models and meta-models.

Model ID	Fold #	MSE sub-model	MSE meta-model
1		$6.227 \cdot 10^{-7}$	$5.998 \cdot 10^{-7}$
2	1	$6.524 \cdot 10^{-7}$	$6.365 \cdot 10^{-7}$
3		$1.716 \cdot 10^{-5}$	$6.613 \cdot 10^{-7}$
4		$6.730 \cdot 10^{-7}$	$5.250 \cdot 10^{-7}$
5	2	$1.415 \cdot 10^{-6}$	$5.353 \cdot 10^{-7}$
6		$5.253 \cdot 10^{-6}$	$5.178 \cdot 10^{-7}$
7		$6.470 \cdot 10^{-7}$	$7.400 \cdot 10^{-7}$
8	3	$3.033 \cdot 10^{-6}$	$7.512 \cdot 10^{-7}$
9		$5.861 \cdot 10^{-7}$	$7.316 \cdot 10^{-7}$
10		$6.292 \cdot 10^{-7}$	$6.186 \cdot 10^{-7}$
11	4	$7.504 \cdot 10^{-7}$	$6.686 \cdot 10^{-7}$
12		$4.537 \cdot 10^{-6}$	$6.747 \cdot 10^{-7}$
13		$8.092 \cdot 10^{-7}$	$6.500 \cdot 10^{-7}$
14	5	$6.292 \cdot 10^{-7}$	$5.996 \cdot 10^{-7}$
15		$7.399 \cdot 10^{-7}$	$6.667 \cdot 10^{-7}$

Table 9
Mean performance values for sub- and meta-models.

	Sub-model	Meta-model	Unit
mean MSE	$2.543 \cdot 10^{-06}$	$6.384 \cdot 10^{-07}$	V^2
std. dev. MSE	$\pm 4.176 \cdot 10^{-06}$	$\pm 7.159 \cdot 10^{-08}$	V^2
mean RMSE	$1.595 \cdot 10^{-3}$	$7.990 \cdot 10^{-4}$	V
std. dev. RMSE	$\pm 2.044 \cdot 10^{-3}$	$\pm 2.676 \cdot 10^{-4}$	V

5.1.1. ANN prediction performance evaluation

In Fig. 11, the results of the nested k-fold cross validation for both the sub-models (a) and the meta-models (b) can be seen. As indicated in Table 4, five folds were used for cross-validation ($k = 5$) and per fold 3 models with the same architecture were trained to be able to evaluate the consistency of the model training procedure and the prediction performance per fold. The comparison of the sub-model and the meta-model performance (c) clearly shows the positive impact of the introduced stacking approach on overall model performance. The mse values for the sub-models within one fold vary by up to two orders of magnitude (fold 1), or at least one order of magnitude (fold 2–4). Only in fold 5 did the sub models exhibit a model consistency comparable to the meta-models. The mse values for the meta-models are all within the same order of magnitude regardless of the fold, leading to a mean mse value of $6.384 \cdot 10^{-07}$ and a standard deviation of $\pm 7.159 \cdot 10^{-08}$. To better classify these values, the square root was taken. The resulting root mean squared error (RMSE) has the same unit as the model target value (V), allowing an easier interpretation of its magnitude. For the meta-models, the mean RMSE of all models concludes at 0.799 mV with a standard deviation of ± 0.268 mV. Compared to the mean sub-model RMSE of 1.595 mV with a standard deviation of ± 2.044 mV, the meta model has an on average 49% lower RMSE with a 86% lower RMSE standard deviation. The individual and mean performance values for both the sub-models and the meta-models are summarized in Tables 8 and 9, respectively.

In Fig. 14, the predictions of the ANN (markers) are compared with the test data held aside during the ANN training (full lines). The three polarization curves with the highest deviation to the test data are highlighted in yellow, grey and blue and their corresponding operation parameters are summarized in Table 10. To identify trends and systematic weaknesses of the trained model, a correlation analysis was performed. Fig. 13 summarizes the results of the correlation analysis of the absolute prediction error and the normalized input variables.

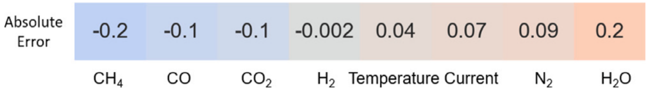

Fig. 13. Correlation of normalized input values with absolute prediction error.

In general, the magnitude of prediction error is only weakly correlated with the input values with maximum correlation coefficients of 0.2 and −0.2 for H_2O and CH_4, respectively. Nevertheless, there is a slight trend towards higher error values for gas compositions with lower amounts of carbon-containing gases and higher amounts of nitrogen and steam. When comparing the three highlighted IV-curves in Fig. 14 with the correlation analysis, only minor trends such as

Table 10
Operation parameters for highlighted IV-curves from Fig. 14.

ID	1	2	3
H_2	75%	15%	5%
H_2O	15%	5%	5%
CO	0%	0%	10%
CO_2	10%	10%	20%
CH_4	8%	2%	4%
N_2	2%	68%	56%
Temperature	725 °C	725 °C	700 °C
Current density	617 $\frac{mA}{cm^2}$	575 $\frac{mA}{cm^2}$	617 $\frac{mA}{cm^2}$
Voltage test data	0.715 V	0.549 V	0.496 V
Voltage predicted	0.712 V	0.561 V	0.499 V
Absolute error	3 mV	12 mV	3 mV

a slight tendency towards lower CO and CO_2 concentrations can be seen. This tendency towards higher voltage deviations at higher current densities, observed at the highlighted IV-curves, is not reflected in the correlation analysis. One possible explanation for this rests with the data generation approach used in this study, an approach which used local refinement in non-linear zones.

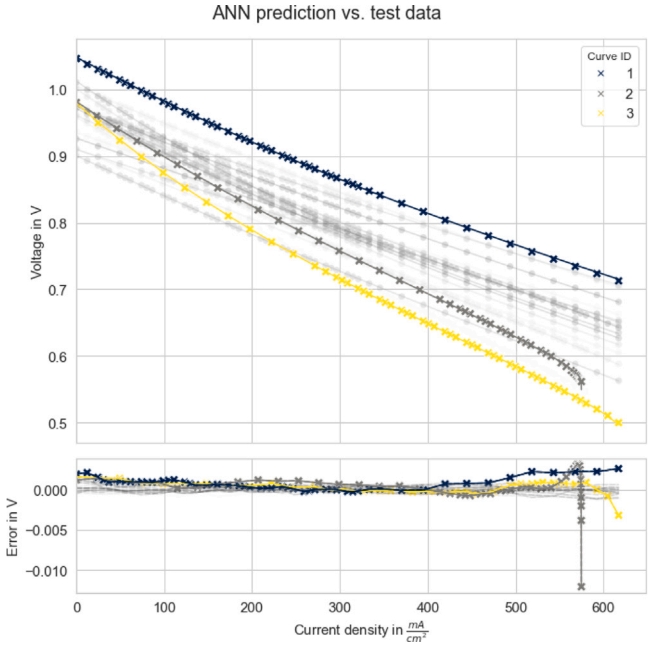

Fig. 14. Comparison of ANN predictions (markers) and final test data (lines). The lines and dots held in light grey indicate IV-curves with a maximum prediction deviation of less than ±2 mV. The three highlighted IV-curves reflect operation modes with a low number of reactive species in the inlet gas composition and thus a high amount of N_2 (Curve ID 3 🟨: N_2: 68%, Curve ID 2 ■: N_2: 56%) and high hydrogen concentrations (Curve ID 1 🟦: H_2: 75%).

The maximum voltage deviation of 3 mV that lies within the operation boundaries (voltage > 0.65 V) results in a maximum relative error of less than 0.47% when assuming the lowest allowed voltage value of 0.65 V.

Compared to literature, where relative errors below 4%–5% for SOFC voltage predictions are commonly stated as satisfactory [6,10,48], the prediction error of the ANN model is one order of magnitude lower. This means that if models with satisfactory prediction accuracy are used as a baseline model for the ANN model, the additional inaccuracy induced by the ANN model has only a minor impact on the overall accuracy.

5.1.2. Validation with measured values and CFD

In addition to the k-fold cross validation, the ANN and the multi-physics model were compared with measured values from an in-house conducted single-cell test and a CFD simulation. As mentioned in Section 2, an anode-supported single cell with 80 cm^2 active surface area and a 300 μm Ni/YSZ anode was studied to validate this. The 2D-CFD simulation was conducted in ANSYS FLUENT 2020 R1 and included the SOFC sub-module with unresolved electrolyte. Additional information on the CFD simulation is provided in [49]. The single cell tests were conducted in a temperature-controlled oven with the ASC cell placed in an alumina cell housing. Polarization curves were measured by means of a BioLogic impedance analyser with an 80 A/3 V booster. Further information regarding the test bench can be found in [50]. The results of this comparison are depicted in Fig. 15.

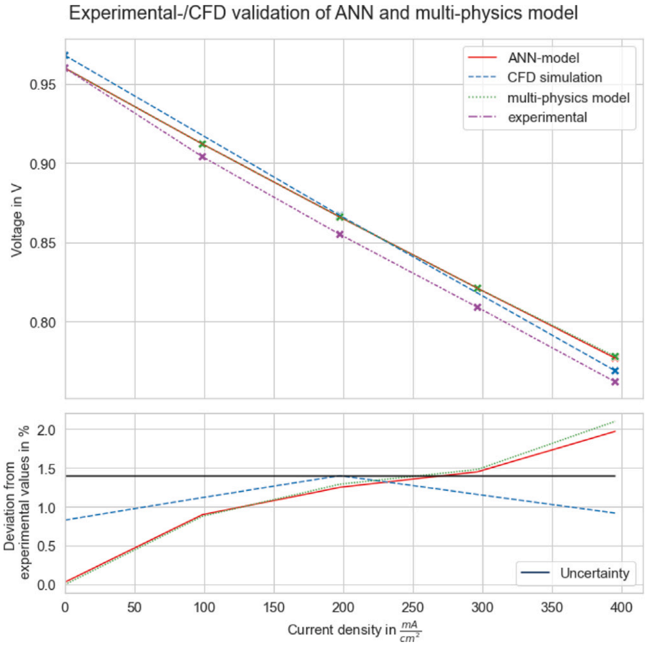

Fig. 15. Comparison of 🟥 ANN predictions, 🟩 multi-physics model and 🟦 CFD simulation results with results from measured values from cell experiments 🟪. The full black line in the lower section reflects the measurement uncertainty of the test rig according to [51].

Due to inevitable measurement uncertainties when conducting experiments for a better classification of the achieved results, a baseline for the measurement uncertainty of the experimental setup of 1.4% was included in Fig. 15 (black full line). The magnitude of the uncertainty was taken from [51], as the evaluated system setup was similar to the one used in this study. For the validation all simulations and the experiment were conducted with the operation parameters listed in Table 11. The voltages of the experiment, the multi-physics model and the ANN model were investigated at 0, 99, 198, 296 and 395 $\frac{mA}{cm^2}$ and at 0, 198 and 395 $\frac{mA}{cm^2}$ for the CFD simulation.

Table 11
Operation parameters used for experimental validation.

Parameter	Value	Unit
H_2	15.6	%
H_2O	14.9	%
CO	15.8	%
CO_2	12.1	%
CH_4	3.9	%
N_2	37.7	%
Mass flow rate fuel	$3.74 \cdot 10^{-5}$	$\frac{kg}{s}$
Mass flow rate air	$4.29 \cdot 10^{-5}$	$\frac{kg}{s}$
Temperature	800	°C
Current density	0–395	$\frac{mA}{cm^2}$

The validation results showed a very good agreement of the CFD-simulation, the multi-physics- and the ANN-model with the experimental data with a voltage deviation below the measurement uncertainty of 1.4% for most values. Only at higher current densities did the multi-physics model and the ANN-model have a slightly higher deviation than the measurement uncertainty Because deviation of the ANN from the multi-physics model is only one mV at maxima, for future research, a promising approach to increase the ANN prediction performance would be to improve the underlying multi-physics model. Especially the implementation of thermal influences should improve the prediction performance significantly according to [13].

5.1.3. Voltage prediction speed

To determine the achievable time reduction required to get the cell voltage of an SOFC for a set of operation parameters, the execution time for one experiment, a 2D-CFD simulation, the multiphysics model and the ANN were compared. Since the time required for the test setup as well as for the pre-processing of the CFD simulation is time consuming and strongly dependent on the skills of personnel, the setup and preparation times were not considered in this comparison. Multiple calculations on 8 parallel cores were assumed for CFD, ANN and multi-physics model resulting in a reduction of the overall time per computation by a factor of 8. In Table 12, the measured execution times are compared and the achievable speed up of the ANN is summarized.

Table 12
Observed execution time and achievable reduction of time required to measure/compute one voltage for a set of operation parameters.

Method	Execution time	Speed-up of ANN
Experiment	45 min	>350 000
2D CFD-simulation	30 min	>200 000
Multi-physics model	3.125 s	>3500
ANN	8.75 ms	–

A speed-up by a factor of more than 350 000 was observed when comparing the ANN model with the experimental setup. Compared to the multi-physics model, that was used for training dataset generation, and the 2D CFD-simulation, the required time for one prediction could be reduced by a factor of more than 3500 and 200 000, respectively.

Closing the prediction accuracy vs. prediction speed gap

In order to evaluate whether the gap between prediction accuracy and prediction speed could be closed within this study, a comparison with values from the literature was additionally conducted and can be seen in Fig. 16.

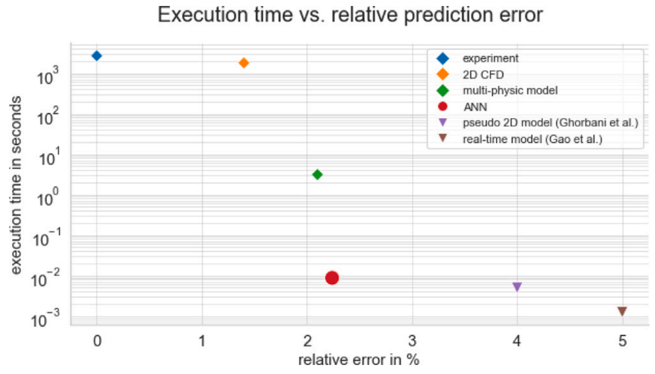

Fig. 16. Comparison of observed execution times and relative voltage prediction errors of this study (left 4 marks) with values from the literature [10,48].

To make the prediction times comparable, the prediction times from the literature sources were rescaled to match with an assumed 8 core CPU setup. The experimental error was defined as 0% as it was used for validating the prediction performance. The relative error values for the CFD, multi-physics model and ANN model were reprinted from Fig. 15. Both literature sources shared a commonality: they only considered hydrogen and steam as fuel. The scope of their work concerned the development of very fast, lightweight SOFC models for online control strategies [10] or real-time control [48]. The slightly slower prediction speed compared to the literature can be explained by two main things. Firstly, the underlying data for the ANN model is of higher complexity as it does not just include hydrogen/steam mixtures as in the two literature sources but also nitrogen and hydrocarbons, along with the additional capability of reflecting the reforming of methane on NI-catalysts. These added features may cause more complex neural network structures and thus more computational resources are required. Secondly, the implemented stacking approach further increases the overall model complexity and with it, the required computational effort.

The achieved prediction accuracy of the ANN allows for the conclusion that it is possible to significantly reduce computation time with only limited compromises regarding accuracy. Even for more complex processes, like internal reforming prediction times close to online and real-time control models can be achieved with this approach.

5.2. Genetic algorithm-process parameter optimization

For the process parameter optimization, the electric power output of the SOFC was defined as an optimization target to be maximized. To be able to evaluate the stability of the optimization approach the genetic algorithm was repeated 10 times. The solutions found are visualized in Fig. 17 and summarized in Table 13. The highest power output found is result 1 with 39.30 W, a fuel composition of 94% H_2, 5% H_2O and 1% N_2 at a temperature of 800 °C and a current density of 598.6 $\frac{mA}{cm^2}$. Results 2–6 are almost identical and have only minor deviations. All solutions found by the GA were similar in that the temperature, the current density and the share of fuel gas in the inlet gas stream were located at the upper limits of the scan range. Thus, gas compositions with just hydrogen as fuel gas are performing slightly better than solutions with a mixture of hydrogen and carbon monoxide (results 7–9) or fuel mixtures containing H_2, CO and CH_4.

The whole optimization process required roughly 1 h on a 12-core Intel Core i9-7920X CPU with 64 Gb of RAM.

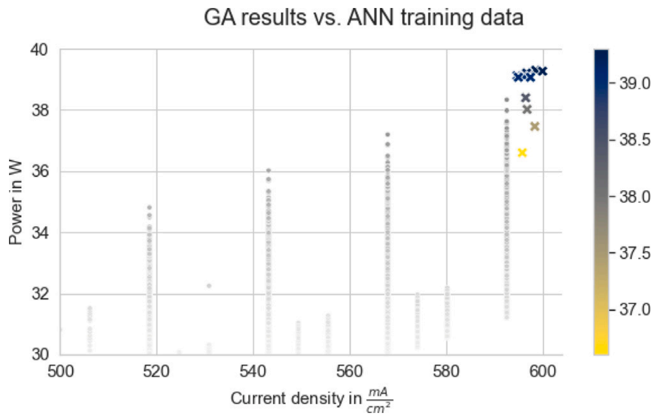

Fig. 17. Results of process parameter optimization with GA. The known data from the ANN training dataset is visualized in the grey dots. The GA results are held in yellow to blue ¨x¨, depending on their value.

5.2.1. Sensitivity analysis

A sensitivity analysis was performed to verify the optimization results from a qualitative point of view. Operation with a fuel mixture consisting of only hydrogen and steam was defined as a scenario for the sensitivity analysis. In order to investigate the effect of a parameter, all parameters listed in Table 14 were left at their mean

Table 13
Results of the genetic algorithm with the optimization target power arranged in descending order.

	GA result										Unit
	1	2	3	4	5	6	7	8	9	10	
H_2	94.0	94.0	94.0	94.0	93.0	93.0	70.0	62.0	48.0	35.0	mol-%
H_2O	5.0	6.0	5.0	5.0	6.0	5.0	5.0	5.0	6.0	15.0	mol-%
CO	0.0	0.0	0.0	0.0	0.0	0.0	25.0	33.0	40.0	40.0	mol-%
CO_2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	mol-%
CH_4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10.0	mol-%
N_2	1.0	0.0	1.0	1.0	1.0	2.0	0.0	0.0	6.0	0.0	mol-%
Temperature	800	799	800	800	800	799	800	799	800	800	°C
Current density	598.6	600.0	596.7	594.7	596.1	595.0	597.5	596.8	598.4	595.8	$\frac{mA}{cm^2}$
Voltage	0.81	0.81	0.81	0.81	0.81	0.81	0.79	0.79	0.77	0.76	V
Power	39.30	39.27	39.21	39.12	39.09	39.07	38.4	38.02	37.46	36.66	W

Table 14
Parameters altered for sensitivity analysis.

Parameter	Min	Mean	Max	Unit
Temperature	700	750	800	°C
Current density	0	300	600	$\frac{mA}{cm^2}$
Reactant/product-ratio at cell inlet	5/95	50/50	95/5	%/%

value with the exception of the parameter under investigation. For comparability, the different parameter values were normalized to a range of ±1. The results of this analysis are shown in Fig. 18. In the left diagram, the entire power range is shown. In the right diagram, the y-axis was adjusted for legibility of the values for temperature and reactant/product-ratio effect. Fig. 18 shows that for maximum power output, all three varied parameters should be kept at their maximum allowable value. Among them, current density has the strongest effect on power output, followed by reactant/product-ratio and temperature. This observed behaviour is consistent with the results suggested by the genetic algorithm. The non-linear range for reactant/product-ratio values below -0.5 can be explained through the effect of fuel starvation that occurs at the mean current density of 300 $\frac{mA}{cm^2}$ and low hydrogen contents in the inlet composition.

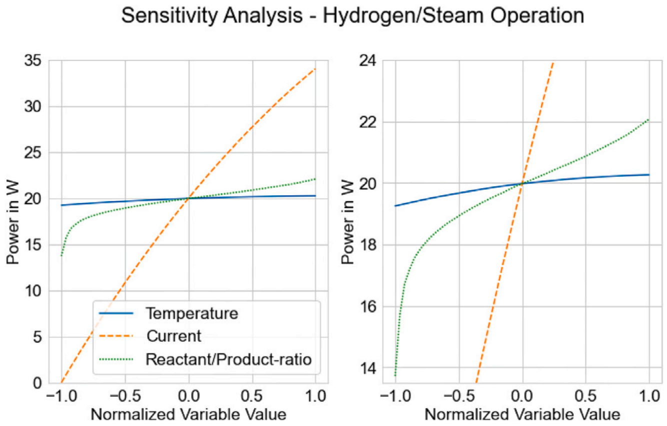

Fig. 18. Sensitivity analysis of SOFC operation with H_2/H_2O as fuel.

6. Conclusions

This work showed that close to real-time, yet accurate, predictions of SOFC voltages by means of an ANN model can be done not just for simple fuel gas compositions containing just H_2/H_2O-mixtures but also for more complex compositions also containing N_2 and the carbonaceous species CO, CO_2 and CH_4. Furthermore, it was demonstrated that coupling an SOFC ANN model with a genetic algorithm is a valid approach for the optimization of operation parameters.

The ANN model was successfully trained with data generated by a multi-physics model to predict the resulting cell voltage for a given inlet fuel composition, cell temperature and current density. Proper hyperparameters for optimal ANN prediction performance were found through algorithm-based tuning. A thorough validation by means of the k-fold cross validation approach revealed an unstable model performance that would not have been detected with just a random split validation. As a countermeasure to this undesirable behaviour, stacking was introduced, leading to a significant increase in both model prediction performance and consistency. The additional experimental validation revealed a good agreement of the ANN-model with measured values from single-cell tests, a CFD-simulation and the underlying multi-physics model. A deviation higher than the measurement uncertainty could only be seen at high current densities. Compared to lightweight SOFC models, lower prediction errors at a comparable prediction speed were achieved despite the higher complexity. The coupling of the developed ANN model with a genetic algorithm was capable of optimizing the operation parameters and thus maximized the electric power output of the single cell. Multiple very good solutions close to the global optimum were suggested by the algorithm, allowing the user to choose from different fuel compositions for the efficient operation of the SOFC. A sensitivity analysis for the most promising operating mode with fuel mixtures containing hydrogen and steam supported the validity of the solutions found by the genetic algorithm. All parameters investigated (current, temperature and reactant/product ratio) led to an increase in the electrical output of the fuel cell when increased.

For future work, more complex optimization goals, such as cell efficiency or multi-objective targets like efficiency maximization and simultaneous power output, will be investigated with this approach. Based on the experiences gained through this work, the following main findings should be taken into account for future work:

- Only data of high quality with a slight focus on regions that are of higher complexity (e.g.: non-linear zones) should be used
- A careful k-fold cross validation is highly recommended to ensure stable, consistent model performance
- The use of algorithm based hyperparameter-tuning and a cyclic learning rate for optimal ANN training has a positive effect on model prediction performance
- Stacking may increase model performance and consistency but at the cost of computational time during model training
- Penalties and boundary conditions for the genetic algorithm should be defined with a gradient dependent on the magnitude of the error in order to allow quicker convergence
- Optimization towards the same goal multiple times might offer alternative yet still viable solutions which would be missed through the use of a single shot optimization

CRediT authorship contribution statement

Felix Mütter: Conceptualization, Methodology, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review

& editing, Visualization. **Clemens Berger:** Conceptualization, Methodology, Formal analysis, Investigation. **Benjamin Königshofer:** Conceptualization, Formal analysis, Writing – review & editing. **Michael Höber:** Conceptualization, Writing – review & editing. **Christoph Hochenauer:** Conceptualization, Resources, Writing – review & editing. **Vanja Subotić:** Conceptualization, Methodology, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

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

Data availability

The data that has been used is confidential.

Acknowledgements

The authors gratefully acknowledge the funding of this project entitled “HyTechonomy - Hydrogen Technologies for Sustainable Economies” (Grant No. 882510) by the Austrian Research Promotion Agency (FFG).

References

[1] Subotić V, Eibl M, Hochenauer C. Artificial intelligence for time-efficient prediction and optimization of solid oxide fuel cell performances. Energy Convers Manage 2021;230:113764. http://dx.doi.org/10.1016/j.enconman.2020.113764, URL http://creativecommons.org/licenses/by/4.0/.

[2] Massardo AF, Lubelli F. Internal reforming solid oxide fuel cell-gas turbine combined cycles (IRSOFC-GT): Part A- Cell model and cycle thermodynamic analysis. J Eng Gas Turbines Power 2000;122(1):27–35. http://dx.doi.org/10.1115/1.483187, URL http://asmedigitalcollection.asme.org/gasturbinespower/article-pdf/122/1/27/5548041/27_1.pdf.

[3] McPhail SJ, Conti B, Kiviaho J. The yellow pages of SOFC technology - International status of SOFC deployment 2017, no. February. 2017, p. 51.

[4] Mitsubishi Hitachi hybrid SOFC-MGTs in trials at Toyota, NGK. Fuel Cells Bull 2017;2017(5):6. http://dx.doi.org/10.1016/S1464-2859(17)30182-7.

[5] Zhang Z, Yue D, Yang G, Chen J, Zheng Y, Miao H, Wang W, Yuan J, Huang N. Three-dimensional CFD modeling of transport phenomena in multi-channel anode-supported planar SOFCs. Int J Heat Mass Transfer 2015;84:942–54. http://dx.doi.org/10.1016/J.IJHEATMASSTRANSFER.2015.01.097.

[6] Badur J, Lemański M, Kowalczyk T, Ziółkowski P, Kornet S. Zero-dimensional robust model of an SOFC with internal reforming for hybrid energy cycles. Energy 2018;158:128–38. http://dx.doi.org/10.1016/J.ENERGY.2018.05.203.

[7] D’Andrea G, Gandiglio M, Lanzini A, Santarelli M. Dynamic model with experimental validation of a biogas-fed SOFC plant. Energy Convers Manage 2017;135:21–34. http://dx.doi.org/10.1016/J.ENCONMAN.2016.12.063.

[8] Kupecki J, Papurello D, Lanzini A, Naumovich Y, Motylinski K, Blesznowski M, Santarelli M. Numerical model of planar anode supported solid oxide fuel cell fed with fuel containing H2S operated in direct internal reforming mode (DIR-SOFC). Appl Energy 2018;230:1573–84. http://dx.doi.org/10.1016/J.APENERGY.2018.09.092.

[9] Kupecki J, Motylinski K, Milewski J. Dynamic analysis of direct internal reforming in a SOFC stack with electrolyte-supported cells using a quasi-1D model. Appl Energy 2018;227:198–205. http://dx.doi.org/10.1016/J.APENERGY.2017.07.122.

[10] Ghorbani B, Vijayaraghavan K. 3D and simplified pseudo-2D modeling of single cell of a high temperature solid oxide fuel cell to be used for online control strategies. Int J Hydrogen Energy 2018;43(20):9733–48. http://dx.doi.org/10.1016/J.IJHYDENE.2018.03.211.

[11] Akkaya AV. Electrochemical model for performance analysis of a tubular SOFC. Int J Energy Res 2007;31(1):79–98. http://dx.doi.org/10.1002/ER.1238, URL https://onlinelibrary.wiley.com/doi/full/10.1002/er.1238 https://onlinelibrary.wiley.com/doi/abs/10.1002/er.1238 https://onlinelibrary.wiley.com/doi/10.1002/er.1238.

[12] Subotić V, Thaller T, Königshofer B, Menzler NH, Bucher E, Egger A, Hochenauer C. Performance assessment of industrial-sized solid oxide cells operated in a reversible mode: Detailed numerical and experimental study. Int J Hydrogen Energy 2020;45(53):29166–85. http://dx.doi.org/10.1016/J.IJHYDENE.2020.07.165.

[13] Tang S, Amiri A, Vijay P, Tadé MO. Development and validation of a computationally efficient pseudo 3D model for planar SOFC integrated with a heating furnace. Chem Eng J 2016;290:252–62. http://dx.doi.org/10.1016/J.CEJ.2016.01.040.

[14] Wang K, Hissel D, Péra MC, Steiner N, Marra D, Sorrentino M, Pianese C, Monteverde M, Cardone P, Saarinen J. A Review on solid oxide fuel cell models. Int J Hydrogen Energy 2011;36(12):7212–28. http://dx.doi.org/10.1016/J.IJHYDENE.2011.03.051.

[15] Marschik C, Roland W, Löw-Baselli B, Steinbichler G. Application of hybrid modeling in polymer processing. 2020.

[16] Arriagada J, Olausson P, Selimovic A. Artificial neural network simulator for SOFC performance prediction. J Power Sources 2002;112(1):54–60. http://dx.doi.org/10.1016/S0378-7753(02)00314-2.

[17] Baldinelli A, Barelli L, Bidini G, Bonucci F, Cansu F, Glu I. Regarding solid oxide fuel cells simulation through artificial intelligence: A neural networks application. 2018, http://dx.doi.org/10.3390/app9010051, URL www.mdpi.com/journal/applsci.

[18] Zhang C, Liu Q, Wu Q, Zheng Y, Zhou J, Tu Z, Chan SH. Modelling of solid oxide electrolyser cell using extreme learning machine. Electrochim Acta 2017;251:137–44. http://dx.doi.org/10.1016/J.ELECTACTA.2017.08.113.

[19] Subotić V, Eibl M, Hochenauer C. Artificial intelligence for time-efficient prediction and optimization of solid oxide fuel cell performances. Energy Convers Manage 2021;230(January). http://dx.doi.org/10.1016/j.enconman.2020.113764.

[20] Deng S, Zhang J, Zhang C, Luo M, Ni M, Li Y, Zeng T. Prediction and optimization of gas distribution quality for high-temperature PEMFC based on data-driven surrogate model. Appl Energy 2022;327. http://dx.doi.org/10.1016/J.APENERGY.2022.120000.

[21] Zeng T, Zhang C, Hu M, Chen Y, Yuan C, Chen J, Zhou A. Modelling and predicting energy consumption of a range extender fuel cell hybrid vehicle. Energy 2018;165:187–97. http://dx.doi.org/10.1016/J.ENERGY.2018.09.086.

[22] Wang Y, Zhang Y, Zhang C, Zhou J, Hu D, Yi F, Fan Z, Zeng T. Genetic algorithm-based fuzzy optimization of energy management strategy for fuel cell vehicles considering driving cycles recognition. Energy 2023;263. http://dx.doi.org/10.1016/J.ENERGY.2022.126112.

[23] Bozorgmehri S, Hamedi M. Modeling and optimization of anode-supported solid oxide fuel cells on cell parameters via artificial neural network and genetic algorithm. 2012, http://dx.doi.org/10.1002/fuce.201100140, URL https://onlinelibrary.wiley.com/doi/10.1002/fuce.201100140.

[24] Yan Z, He A, Hara S, Shikazono N. Modeling of solid oxide fuel cell (SOFC) electrodes from fabrication to operation: Microstructure optimization via artificial neural networks and multi-objective genetic algorithms. Energy Convers Manage 2019;198. http://dx.doi.org/10.1016/J.ENCONMAN.2019.111916.

[25] Le MV, Nguyen TA, Nguyen TAN. Modeling and optimization of the BSCF-based single-chamber solid oxide fuel cell by artificial neural network and genetic algorithm. J Chem 2019;2019. http://dx.doi.org/10.1155/2019/7828019.

[26] Xu H, Ma J, Tan P, Chen B, Wu Z, Zhang Y, Wang H, Xuan J, Ni M. Towards online optimisation of solid oxide fuel cell performance: Combining deep learning with multi-physics simulation. Energy AI 2020;1. http://dx.doi.org/10.1016/J.EGYAI.2020.100003.

[27] Krstajic D, Buturovic LJ, Leahy DE, Thomas S. Cross-validation pitfalls when selecting and assessing regression and classification models. J Cheminform 2014;6(1):1–15. http://dx.doi.org/10.1186/1758-2946-6-10.

[28] Maier L, Schädel B, Delgado KH, Tischer S, Deutschmann O. Steam reforming of methane over nickel: Development of a multi-step surface reaction mechanism. Top Catalysis 2011;54(13–15):845–58. http://dx.doi.org/10.1007/S11244-011-9702-1/FIGURES/10, URL https://link.springer.com/article/10.1007/s11244-011-9702-1.

[29] Davini D, Samineni B, Thomas B, Tran AH, Zhu C, Ha K, Dasika G, White L. Using physics-informed regularization to improve extrapolation capabilities of neural networks. 2021.

[30] Subotić V, Schluckner C, Hochenauer C. An experimental and numerical study of performance of large planar ESC-SOFCs and experimental investigation of carbon depositions. J Energy Inst 2016;89(1):121–37. http://dx.doi.org/10.1016/j.joei.2015.01.004.

[31] Neidhardt JP, Bessler WG. Microkinetic modeling of nickel oxidation in solid oxide cells: Prediction of safe operating conditions. Chem-Ing-Tech 2019;91(6):843–55. http://dx.doi.org/10.1002/CITE.201800197, URL https://onlinelibrary.wiley.com/doi/10.1002/cite.201800197.

[32] Li J. Oxygen evolution reaction in energy conversion and storage: Design strategies under and beyond the energy scaling relationship. Nano-Micro Lett 2022;14(1):3. http://dx.doi.org/10.1007/s40820-022-00857-x.

[33] Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X. TensorFlow: Large-scale machine learning on heterogeneous systems. 2015, Software available from tensorflow.org. URL https://www.tensorflow.org/.

[34] Chollet F, et al. Keras. 2015, URL https://github.com/fchollet/keras.

[35] O'Malley T, Bursztein E, Long J, Chollet F, Jin H, Invernizzi L, et al. KerasTuner. 2019, https://github.com/keras-team/keras-tuner.

[36] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: Machine learning in Python. J Mach Learn Res 2011;12:2825–30.

[37] Smith LN. Cyclical learning rates for training neural networks. In: Proceedings - 2017 IEEE winter conference on applications of computer vision, WACV 2017 (April). 2017, p. 464–72. http://dx.doi.org/10.1109/WACV.2017.58, arXiv:1506.01186.

[38] Klambauer G, Unterthiner T, Mayr A, Hochreiter S. Self-normalizing neural networks. Adv Neural Inf Process Syst 2017;2017-Decem:972–81, arXiv:1706.02515.

[39] Li L, Jamieson K, Rostamizadeh A, Talwalkar A. Hyperband: A novel bandit-based approach to hyperparameter optimization. Tech. rep., 2018, p. 1–52, URL http://jmlr.org/papers/v18/16-558.html.

[40] Brochu E, Cora VM, de Freitas N. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. 2010, arXiv:1012.2599. URL http://arxiv.org/abs/1012.2599.

[41] Frazier PI. A tutorial on Bayesian optimization. Tech. rep., 2018, arXiv:1807.02811v1.

[42] Kim Y, Chung M. An approach to hyperparameter optimization for the objective function in machine learning. Electronics 2019;8(11):1267. http://dx.doi.org/10.3390/electronics8111267, URL https://www.mdpi.com/2079-9292/8/11/1267.

[43] Ramachandran P, Zoph B, Le Google Brain QV. SWISH: A self-gated activation function. Tech. rep., 2017, arXiv:1710.05941v1.

[44] Lu L, Shin Y, Su Y, Karniadakis GE. Dying ReLU and initialization: Theory and numerical examples. Commun Comput Phys 2019;28(5):1671–706. http://dx.doi.org/10.4208/cicp.OA-2020-0165, arXiv:1903.06733. URL http://arxiv.org/abs/1903.06733.

[45] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting. J Mach Learn Res 2014. http://dx.doi.org/10.1016/0370-2693(93)90272-J.

[46] Weicker K. Evolutionäre algorithmen. Springer Fachmedien Wiesbaden; 2015, http://dx.doi.org/10.1007/978-3-658-09958-9.

[47] Solgi R. Python package for genetic algorithm (GA). 2020, http://dx.doi.org/10.5281/zenodo.3784414.

[48] Gao F, Simoes MG, Blunier B, Miraoui A. Development of a Quasi 2-D modeling of tubular solid-oxide fuel cell for real-time control. IEEE Trans Energy Convers 2014;29(1):9–19. http://dx.doi.org/10.1109/TEC.2013.2293223.

[49] Pongratz G, Subotić V, Hochenauer C, Scharler R, Anca-Couce A. Solid oxide fuel cell operation with biomass gasification product gases: Performance- and carbon deposition risk evaluation via a CFD modelling approach. Energy 2022;244. http://dx.doi.org/10.1016/J.ENERGY.2021.123085.

[50] Stoeckl B, Subotić V, Preininger M, Schroettner H, Hochenauer C. SOFC operation with carbon oxides: Experimental analysis of performance and degradation. Electrochim Acta 2018;275:256–64. http://dx.doi.org/10.1016/J.ELECTACTA.2018.04.036.

[51] Momma A, Takano K, Tanaka Y, Kato T. Evaluating uncertainty for the standardization of single cell/stack power generation performance tests for SOFC. Synth Engl Ed 2013;5(4):251–61. http://dx.doi.org/10.5571/syntheng.5.251.