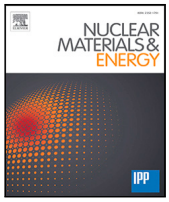
Nuclear Materials and Energy 34 (2023) 101396

Contents lists available at ScienceDirect

Nuclear Materials and Energy

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



Towards fast surrogate models for interpolation of tokamak edge plasmas

Stefan Dasbach *, Sven Wiesen

Forschungszentrum Jülich GmbH, Institut für Energie- und Klimaforschung - Plasmaphysik, Jülich, 52425, Germany

ARTICLE INFO

Keywords:
Solps
Plasma exhaust
Divertor
Surrogate
Machine learning
Neural network

ABSTRACT

One of the major design limitations for tokamak fusion reactors is the heat load that can be sustained by the materials at the divertor target. Developing a full understanding of how machine or operation parameters affect the conditions at the divertor requires an enormous number of simulations. A promising approach to circumvent this is to use machine learning models trained on simulation data as surrogate models. Once trained such surrogate models can make fast predictions for any scenario in the design parameter space. In future such simulation based surrogate models could be used in system codes for rapid design studies of future fusion power plants. This work presents the first steps towards the development of such surrogate models for plasma exhaust and the datasets required for their training. Machine learning models like neural networks usually require several thousand data points for training, but the exact amount of data required varies from case to case. Due to the long runtimes of simulations we aim at finding the minimal amount of training data required. A preliminary dataset based on SOLPS-ITER simulations with varying tokamak design parameters, including the major radius, magnetic field strength and neutral density is constructed. To be able to generate more training data within reasonable computation time the simulations in the dataset use fluid neutral simulations and no fluid drift effects. The dataset is used to train a simple neural network and Gradient Boosted Regression Trees and test how the performance depends on the number of training simulations.

1. Introduction

The sustainable heat load at the divertor targets poses a major design limitation for tokamak fusion reactors. In divertor tokamaks the core plasma is surrounded by a region of magnetic field lines which start and end at the divertor targets. This so called scrape-off layer (SOL) is crucial for the transport of heat and particles from the main plasma to the divertor. For the development of future reactors it is vital to have models that accurately describe the effects of reactor design and operation on the scrape-off layer and the heat load at the divertor. Plasma exhaust codes like SOLPS-ITER simulate the multi-physics phenomena in the SOL sufficiently [1], but each simulation yields only a result for a single tested scenario. Due to the many relevant design and operating parameters and the long simulation run times it is impossible to simulate all reasonable scenarios and some simulation parameters (e.g. anomalous transport) are unknown for new reactors. This gap can be bridged by surrogate models. Surrogate models are machine learning models trained on simulation data, that are than able to interpolate in the parameter space and produce fast results for all scenarios. Because each datapoint for training requires a computationally demanding simulation, the goal in surrogate modeling is to train a model with the least amount of data possible. While this approach still requires running many simulations as training data for the surrogate model, the overall computational effort should be less than exploring the high-dimensional parameter space by simulation alone. In addition some surrogate models like neural networks are differentiable and can therefore be used for gradient based design optimization. Surrogate modeling is being used in a vast number of fields, e.g. engineering [2], medicine [3], neuroscience [4], plasma physics [5] and has also been tested for scrape-off layer simulations [6–8].

We aim at testing the creation of simulation datasets and surrogate models covering the influence of reactor design and operation on the dynamics in the scrape-off layer.

2. Methods

2.1. Data generation

In this study we run a large number of scrape-of layer plasma simulations while systematically varying eight tokamak parameters. The tokamak geometry is lower single null with vertical targets. The simulations are conducted with the SOLPS-ITER code package [9]. From this code only the fluid solver B2.5 is used [10]. This code solves the Braginskii equations for the plasma ions and similar fluid

* Corresponding author.
E-mail address: s.dasbach@fz-juelich.de (S. Dasbach).

https://doi.org/10.1016/j.nme.2023.101396
Received 29 June 2022; Received in revised form 3 February 2023; Accepted 22 February 2023
Available online 25 February 2023

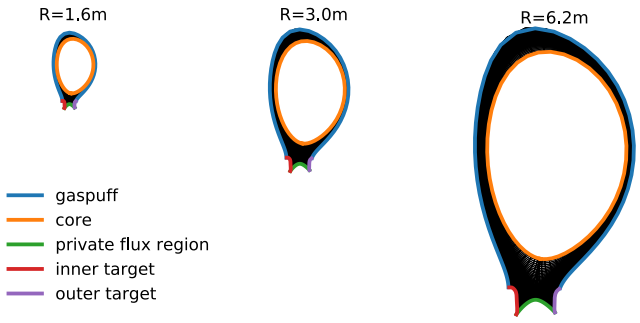


Fig. 1. Computational grids of 3 simulations with varying major radius R. The colors mark grid borders used for boundary conditions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

equations for neutral gas particles. This fluid neutral model is less computationally demanding than the more exact kinetic neutral model and therefore allows for running more simulations. While this approach might reduce the quantitative accuracy of the simulations, the overall trends are still recovered [11] such that the lessons learned can be transferred to a repetition of this study using kinetic neutral models. The simulations contain eight plasma fluids, deuterium ions and all ionization stages of nitrogen and two neutral fluids, atomic deuterium and nitrogen. The simulations shown here are all conducted using similar settings as in [12], that showed to yield stable simulations in low and high density regimes. All simulations are started from a state with constant densities ($\left[n_D, n_{D^+}, n_N, n_{N^{x+}}\right] = \left[10^{16}, 10^{19}, 10^{12}, 10^{12}\right]\ 1/\mathrm{m}^3$) and temperatures ($T_e = T_i = 100\,\mathrm{eV}$) in the whole simulation domain. To scan different operational tokamak regimes we vary the tokamak size R, magnetic field strength B, deuterium gas puff D_{puff}, nitrogen gas puff N_{puff}, deuterium core fueling $Dcore$, input power P_{in}, and the cross field diffusivities for the ion densities $D_\perp$ and the anomalous thermal diffusivity χ. Varying tokamak size and magnetic field requires changes to the computational grid, which is aligned to the magnetic field lines. The currently employed grid generators require manual corrections (or at least manual oversight). Because of the many runs, it is impossible to inspect all grids manually. We therefore restrict ourselves to variations that keep the shape of the computational grid unchanged. Variations in the major radius of the tokamak result in a proportional stretching of the x and y coordinates of all grid cells and the poloidal magnetic field strength is changed proportionally to the toroidal magnetic field strength in each grid cell. Similar scalings were already used in [13]. Examples of the resulting computational grids are depicted in Fig. 1. The computational grid has 102 cells in the poloidal and 48 cells in the radial direction. The number of grid cells stays constant for all tokamak sizes. Because the number of grid cells is higher than usually used for SOLPS simulations and simulations with fluid neutrals are resilient against low grid resolutions [14] we assume that our grid resolution is adequate for all tested tokamak sizes.

While R and B determine the size and the magnetic field strength inside the computational grid, D_{puff}, N_{puff}, D_{core}, P_{in} act as boundary conditions for the solved plasma equations. D_{puff} and N_{puff} determine the source of neutral atoms at the outside boundary of the grid, D_{core} determines the source of deuterium ions at the core boundary and P_{in} determines the energy flowing into the simulation domain from the core (see Fig. 1). At all these boundaries the total sources are distributed uniformly along the given boundary. The power P_{in} is distributed evenly to the electron and ion heat balance equations. The domain boundary in the private flux region acts as pump for all particles, with fixed pumping rates. The pumped flux for each fluid is given by $\Gamma_a = r_a c_{s,a} n_a$ with $c_{s,a}$ and n_a being the sound speed and density of the fluid and the constant pump rates $r_a = 0.5$ for neutrals and $r_a = 0.001$ for ions. $D_\perp$ and χ are scalar parameters in the transport equations and are constant on the whole grid. The

Table 1
Overview of the parameters varied in the dataset. For each varied parameter the minimal and maximal values and the units are given. The scale describes whether the datapoints are distributed uniformly over a linear scale between minimum and maximum (lin) or uniformly over a logarithmic scale between minimum and maximum (log).

	R	B	D_{puff}	N_{puff}	D_{core}	P_{in}	$D_\perp$	χ
min	1	1	10^{20}	10^{18}	10^{19}	10	0.1	0.1
max	10	10	10^{24}	10^{23}	10^{24}	200	2	2
Units	m	T	at/s	at/s	at/s	MW	m^2/s	m^2/s
Scale	lin	lin	log	log	log	lin	lin	lin

parameters for each simulation are selected inside the given limits (Table 1) according to a Sobol sequence [15]. This low-discrepancy sequence covers the parameter space more efficiently compared to random numbers or other such sequences [16]. For D_{puff}, N_{puff}, D_{core} the datapoints are distributed logarithmically (base 10) in the given ranges. The simulations are run for intervals of 1 s simulated time with a timestep of 10^{-5} s until the simulation is either converged to a steady state, shows (stable) oscillations or has been run for a total of 3 s. Simulations are defined as converged when the ratio between the maximum and the minimum outer midplane electron separatrix density in the last 0.1 s of a 1 s run is less than 0.1% and a simulation counts as oscillating if the ratio between the maximum/minimum outer midplane separatrix density in the first and last 0.1 s of a 1 s run is less than 0.5% (similar as in [12]). A total of 4096 training simulations are conducted using this scheme. To act as a test set an additional 1024 simulations are conducted similarly with the exception of being distributed pseudo-randomly in the parameter space.

2.2. Surrogate training

The final values of all stable (non diverging) simulations are used for training of surrogate models. This introduces the simulation time as hidden parameter to our surrogate models since not all simulations reached a steady state, but since all non converged simulations are run equally long this should not impact the surrogate performance immensely. The simulation data is used to train fully-connected feed-forward neural networks (NN) [17] and gradient boosted regression trees (GBRT) [18]. Both model types receive as input the eight simulation parameters (scaled to mean zero and standard deviation one) and are trained to predict the electron temperature in the last row of grid cells in front of the outer divertor target (output dimensionality: 48). In the case of GBRT this means training 48 models in parallel, each predicting the scalar temperature in its designated grid cell. Before training the temperatures are normalized to standard gaussians using a quantile transformer [19] with 100 quantiles (independently for each target grid cell). Such a non-linear transformation is required due to the large range of temperatures present in the simulations.

For both model types a hyperparameter search was conducted to find the best performing models. For GBRT a small grid search and for the NN a random search with 100 trials were run. The respective searched parameter ranges are depicted in Tables 2 and 3. All NN were trained with mean absolute error loss, the Adam optimizer [20] and Early-Stopping based on the validation loss with a patience of 100 epochs (patience scaled proportionally when training on data subset). In both cases the models were trained with 5-fold cross validation and the best model was selected based on the lowest mean validation score across the 5 folds. To evaluate the performance on the test set a final model was constructed by retraining the GBRT with the best hyperparameters from scratch on all the training data (of the selected subset). Instead of retraining the NN with the best hyperparameters, the final predictions on the test set were generated by averaging the predictions from the 5 networks trained on the different folds of the training data. The results in Section 3.2 always depict the performance on the test set of the final models. It is vital to find surrogate models

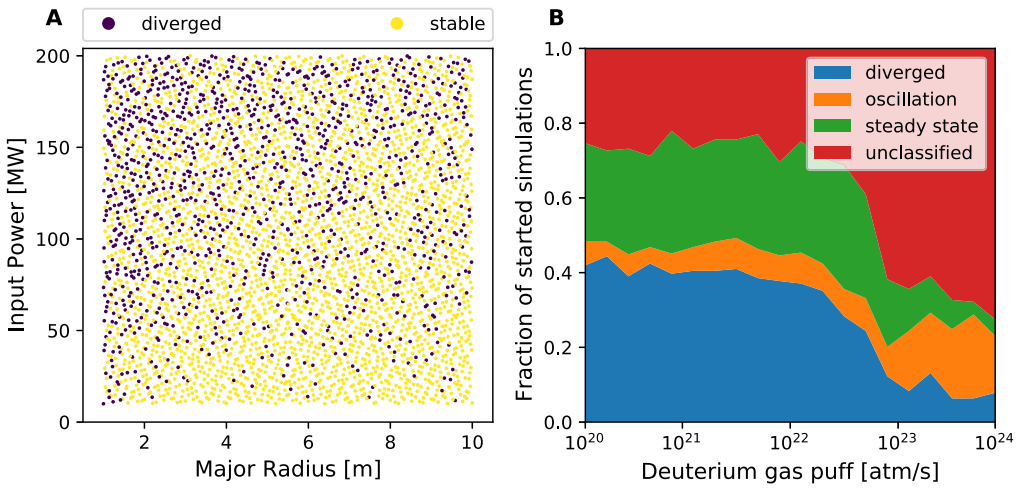


Fig. 2. Status of all started training simulations. A Each dot represents a started simulation with a given input power P_{in} and major radius R. The color represents whether the simulation diverged. B The fraction of all started simulations per deuterium gas puff D_{puff} that are categorized into one of the 4 states (diverged, oscillation, steady state, unclassified). The classification into oscillation and steady state is described in Section 2.1 and a simulation counts as unclassified if it reaches the full 3 s runtime without fitting one of the other categories. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
GBRT grid search - All parameter combinations are tested.

Parameter	Tested values
Tree depth	{3,4,5,6}
Learning rate	{0.01,0.05,0.1}
Number of estimators	[200,1600] in steps of 200

Table 3
NN random search - 100 trials with randomly chosen parameters from the given ranges. The learning rate and L2 regularization were sampled in logarithmic domain. Batch normalization was only tested with Relu and Elu activations. Each hidden layer has the same number of neurons.

Parameter	Searched range
Number of hidden layers	[1,10]
Neurons per hidden layer	[100,2000]
Learning rate	[0.0001,0.005]
L2 regularization	[0.0001,0.01]
Activation	{Relu,Elu,Selu}
Mini-batch size	[20,200]
Batch normalization	{True,False}

that can be trained with minimal amounts of training data, to be able to create future surrogate models on more computationally demanding simulations. To test our model types in that regard we repeated the whole procedure of parameter search based on cross validation and re-training/averaging of a final model independently for different amounts of simulations. Because the simulations are generated according to a sobol sequence we do not select random subsets of the data but the first $2^N \quad N \in \{7, 8, 9, 10, 11, 12\}$ simulations. Due to the characteristics of the sobol sequence this ensures that also for the smaller simulation counts the whole parameter space is explored. Because the sobol sequence determines the started simulations, and some of these simulations diverge, the actual number of stable simulations used is slightly smaller (see Fig. 4). The test set to report the final performances stays always the same.

2.3. Software and hardware

All simulations are conducted on the batch nodes of JURECA DC [21] with SOLPS-ITER version 3.0.7 [9,13]. The analysis and surrogate training is done with Python version 3.8.5 and the packages numpy 1.18.5 [22], scipy 1.5.2 [23], pandas 1.1.3 [24,25], matplotlib 3.3.1 [26], scikit-learn 0.23.2 [19] and tensorflow 2.3.1 [27].

3. Results and discussion

3.1. Overview of the dataset

The parameters selected according to the sobol sequence fill the whole parameter space with constant density (Fig. 2A). Of all 4096 started training simulations 1198 diverge, meaning a state variable increases to unphysical high values, causing a code error. As is depicted in Fig. 2A these diverging simulations occur in all regions of the parameter space. This happens because the independent variation of the eight simulation parameters, leads to unsensible combinations of parameters which describe systems of equations with no solution. It can however not be concluded whether all of the diverging simulations describe unphysical situations or whether some could be solved with a different numerical procedure. Not all simulations are equally likely to diverge but e.g. more simulations diverge with small tokamak size and large input power (Fig. 2A) or with low deuterium gas puffs (Fig. 2B). The simulations that do not diverge can be further categorized into ones converging to a steady state or exhibiting stable oscillations (see Section 2). Such oscillations have previously been seen in similar fluid neutral simulations where the oscillations were deemed to have a physical origin [28]. This is further substantiated by studies that identified physical origin mechanisms of oscillations in the scrape-off layer [29,30]. For small deuterium gas puffs more simulations converge to a steady state than oscillations, while this is reversed for high deuterium gas puffs (Fig. 2B). Due to the reduced divergence rate, we have more stable simulations with high deuterium gas puff. Of these stable simulations a higher fraction reaches neither of the converged states during the 3 s runtime (Fig. 2B). This is most likely because the simulations with high deuterium gas puff tend to have higher plasma densities and because our initial state has a very low plasma density, these runs require longer to reach a final state. Depending on the fraction of the simulations that will then end in the static convergence compared to oscillations, this could also change the before described ratio. The simulations cover a diverse field of physical regimes, distinguishable by the present temperature gradients. In 46% of all stable simulations the separatrix electron temperature at the outer midplane and the outer target differ by less than 20% suggesting that these are in a sheath-limited regime, meaning the only meaningful temperature gradient occurs in the sheath, which is not modeled in these simulations (only as a boundary condition). 13% of the simulations have a temperature gradient towards the inner target, while the temperature at the outer target is still close to the outer

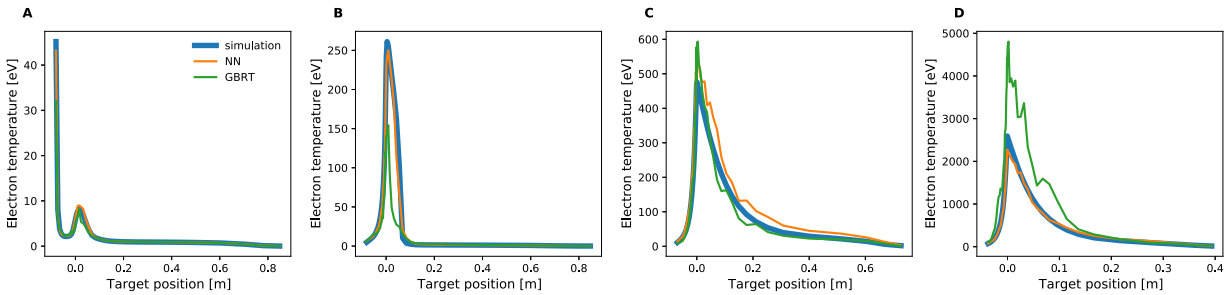


Fig. 3. Predictions of the trained surrogate models on the test data. A,B,C,D show the outer divertor target temperatures of 4 test simulations and the predictions of the final surrogate models trained on all training simulations.

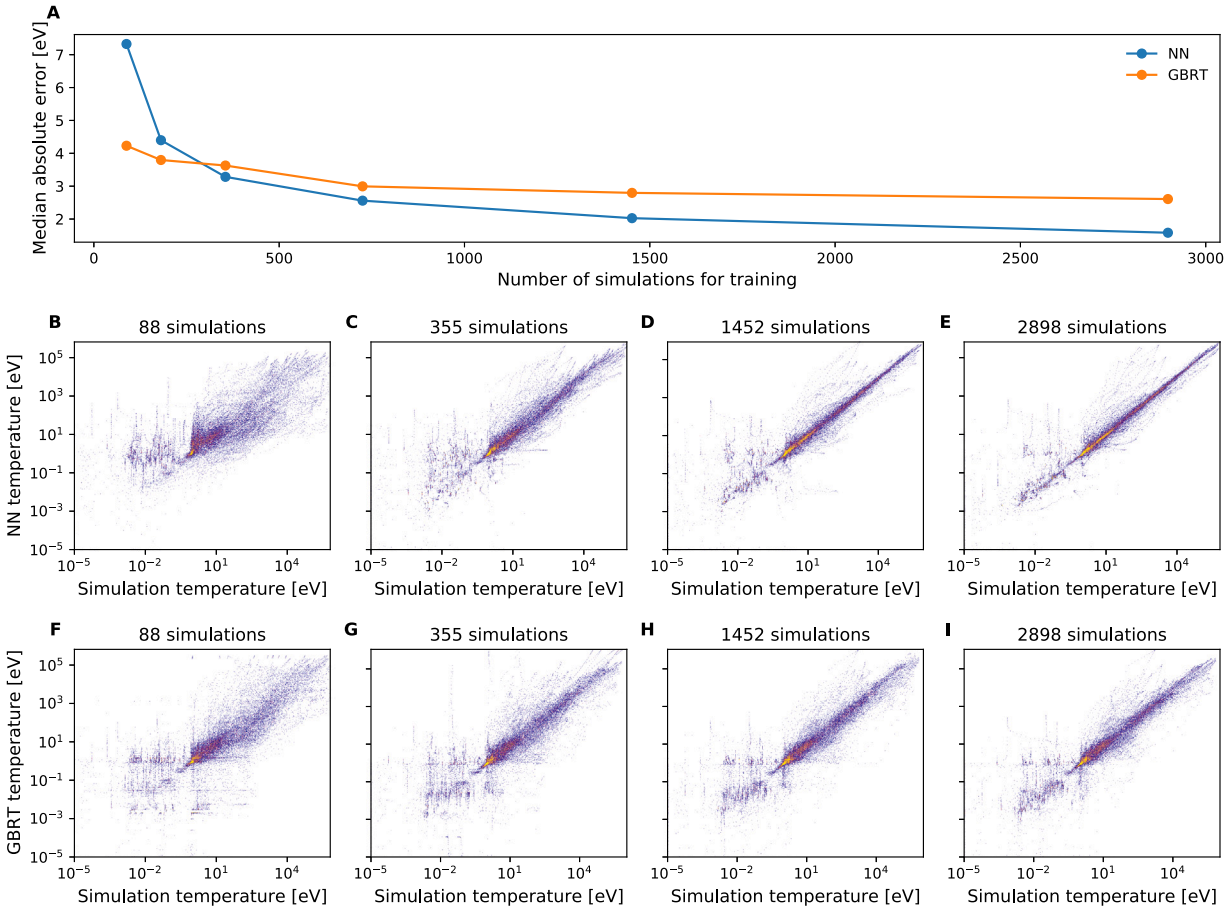


Fig. 4. Performance of surrogate models trained on data subsets. A The median absolute error on the test set for NN and GBRT models with optimized hyperparameters trained with different amounts of training data. B-I show the predicted temperatures of all 48 target grid cells against the respective temperatures in the simulations for all simulations in the test set. The color gets brighter with more points at the same plot position. B-E show the predictions of the NN models and F-I of the GBRT models. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

midplane, while 35% of the simulations have significant temperature gradients towards both divertor targets. In an additional 5% of the simulations the whole simulation domain is cooled down, such that the outer midplane temperature is below 10 eV. Due to these different regimes the temperature values at the outer divertor target used for training the surrogate span several orders of magnitudes. The lowest target temperature is 10^{-5} eV, the highest 10^{6} eV, while the mean is 4822 eV with a standard deviation of 31 570 eV, the median target temperature is 6.5 eV.

3.2. Training a surrogate model

The predictions of the surrogate models developed on the whole training data for four test simulations are depicted in Fig. 3. Both the GBRT and NN model predict always temperatures of the correct order of magnitude, which is already an achievement, given the vast temperature ranges present in the data. Also the general shape of the temperature profiles across the target is successfully captured by both models. Especially the spiking temperature at the left side of the simulation in Fig. 3A is recovered by both models although the maximum temperature in the GBRT model is 10 eV too low. These temperature spikes are present in several simulations with low target temperatures. We believe these are simulation artifacts stemming from the close proximity of these target grid cells to the grid boundary in the private flux region (see Fig. 1) which acts as particle pump in the fluid neutral simulations. However a full analysis of this phenomena is still required. In the examples of Fig. 3B and D the GBRT model predicts some temperatures by a factor of 2–3 too high or too low. The NN

predictions correspond very accurately to the simulation results, only Fig. 3C shows some stronger discrepancies. Further it can be observed that the smoothness of the profiles in the simulations is not necessarily reproduced by the models (Fig. 3C and D). To generate more realistic profiles it could therefore prove useful to apply some smoothing functions after the model predictions or constrain the models to provide smooth profiles themselves.

Instead of evaluating the surrogate accuracy solely on a few examples Fig. 4 compares the predictions with the simulations on the whole test set. Comparing Fig. 4E to I shows that indeed the predictions of the NN are closer to the simulations than the GBRT predictions. Nevertheless in some cases also the temperatures predicted by the neural network can be orders of magnitude too high or too low. Interestingly Fig. 4E and I show some similar patterns in the most extreme too high predicted temperatures, so it is likely these stem from the same test simulations. With smaller number of training simulations the accuracy of the neural network decreases. While in Fig. 4C the network is still able to recover at least the order of magnitude in the high temperature range, below 100 training simulations the predictions become a lot more diffuse (Fig. 4B) and show only slight correlation to the true simulation temperatures. Fig. 4F-I show similar results for the GBRT models. To quantify the performance of the models trained with different amounts of data we calculate the median absolute error on the fixed test set (Fig. 4A). Because the normalization with the quantile transformer has to be redone for each training data set to avoid data spillage, the prediction error is evaluated after back-transformation into the unnormalized domain. The test error shows a large improvement with the number of training simulations in case of the neural network based models and a smaller improvement in case of the GBRT models. For smaller number of simulations the GBRT models achieve a better score while for high number of simulations the neural networks seem superior. The best median test error of the neural networks is 1.6 eV and 2.6 eV of the GBRT model. However since we employed only a very coarse optimization for few of the hyperparameters in the GBRT models, future work with a more thorough optimization might be able to improve GBRT performance. Similarly also for neural network based surrogates different architectures are worth investigating, such as the use of convolutional layers that benefit from the correlation between temperature values of neighboring grid cells. Since some of the simulations in this dataset show unreasonable conditions with e.g. extremely high or low plasma core temperatures it should also be tested if an exclusion of those improves the surrogate accuracy in the reactor relevant regimes. An interesting question is how models behave if they are given more input information like the upstream conditions or if models can be informed by analytically constructed reduced models (e.g. the 2-point model [31]). Finally future models should predict not only the temperatures but also densities and particle and heat fluxes and make these predictions not only at the divertor target but in the whole simulation domain.

CRediT authorship contribution statement

Stefan Dasbach: Conceptualization, Methodology, Software, Investigation, Data curation, Visualization, Writing – original draft. **Sven Wiesen:** Conceptualization, Writing – review & editing, Supervision.

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

Data will be made available on request.

Acknowledgments

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

The authors gratefully acknowledge the computing time granted by the JARA Vergabegremium and provided on the JARA Partition part of the supercomputer JURECA at Forschungszentrum Jülich.

References

[1] A.S. Kukushkin, H.D. Pacher, V. Kotov, G.W. Pacher, D. Reiter, Finalizing the ITER divertor design: The key role of SOLPS modeling, Fusion Eng. Des. 86 (12) (2011) 2865–2873, http://dx.doi.org/10.1016/j.fusengdes.2011.06.009, URL https://www.sciencedirect.com/science/article/pii/S092037961100514X.

[2] G. Sun, S. Wang, A review of the artificial neural network surrogate modeling in aerodynamic design, Proc. Inst. Mech. Eng. G 233 (16) (2019) 5863–5872, http://dx.doi.org/10.1177/0954410019864485, Publisher: IMECHE.

[3] W.A. Pruett, R.L. Hester, The Creation of Surrogate Models for Fast Estimation of Complex Model Outcomes, PLoS One 11 (6) (2016) e0156574, http://dx.doi.org/10.1371/journal.pone.0156574, URL https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0156574. Publisher: Public Library of Science.

[4] Y. Zhang, L.-S. Young, DNN-assisted statistical analysis of a model of local cortical circuits, Sci. Rep. 10 (1) (2020) 20139, http://dx.doi.org/10.1038/s41598-020-76770-3, URL https://www.nature.com/articles/s41598-020-76770-3. Publisher: Nature Publishing Group.

[5] K.L. van de Plassche, J. Citrin, C. Bourdelle, Y. Camenen, F.J. Casson, V.I. Dagnelie, F. Felici, A. Ho, S. Van Mulders, Fast modeling of turbulent transport in fusion plasmas using neural networks, Phys. Plasmas 27 (2) (2020) 022310, http://dx.doi.org/10.1063/1.5134126, URL https://aip.scitation.org/doi/10.1063/1.5134126. Publisher: American Institute of Physics.

[6] R. Preuss, U.v. Toussaint, Gaussian Processes for SOLPS Data Emulation, Fusion Sci. Technol. (2017) http://dx.doi.org/10.13182/FST15-178, URL https://www.tandfonline.com/doi/pdf/10.13182/FST15-178?needAccess=true. Publisher: Taylor & Francis.

[7] V. Gopakumar, D. Samaddar, Image mapping the temporal evolution of edge characteristics in tokamaks using neural networks, Vol. 1, IOP Publishing, 2020, 015006, http://dx.doi.org/10.1088/2632-2153/ab5639.

[8] B. Zhu, M. Zhao, H. Bhatia, X.-q. Xu, P.-T. Bremer, W. Meyer, N. Li, T. Rognlien, Data-driven model for divertor plasma detachment prediction, J. Plasma Phys. 88 (5) (2022) 895880504, http://dx.doi.org/10.1017/S002237782200085X, URL https://www.cambridge.org/core/journals/journal-of-plasma-physics/article/datadriven-model-for-divertor-plasma-detachment-prediction/BC221B0DF5A8857B08AE74475C413243. Publisher: Cambridge University Press.

[9] S. Wiesen, D. Reiter, V. Kotov, M. Baelmans, W. Dekeyser, A.S. Kukushkin, S.W. Lisgo, R.A. Pitts, V. Rozhansky, G. Saibene, I. Veselova, S. Voskoboynikov, The new SOLPS-ITER code package, J. Nucl. Mater. 463 (2015) 480–484, http://dx.doi.org/10.1016/j.jnucmat.2014.10.012, URL http://www.sciencedirect.com/science/article/pii/S0022311514006965.

[10] V.A. Rozhansky, S.P. Voskoboynikov, E.G. Kaveeva, D.P. Coster, R. Schneider, Simulation of tokamak edge plasma including self-consistent electric fields, Nucl. Fusion 41 (4) (2001) 387–401, http://dx.doi.org/10.1088/0029-5515/41/4/305, URL https://doi.org/10.1088/0029-5515/41/4/305. Publisher: IOP Publishing.

[11] D.P. Coster, Detachment physics in SOLPS simulations, J. Nucl. Mater. 415 (1, Supplement) (2011) S545–S548, http://dx.doi.org/10.1016/j.jnucmat.2010.12.223, URL http://www.sciencedirect.com/science/article/pii/S0022311510010561.

[12] D.P. Coster, Exploring the edge operating space of fusion reactors using reduced physics models, Nucl. Mater. Energy 12 (2017) 1055–1060, http://dx.doi.org/10.1016/j.nme.2016.12.033, URL http://www.sciencedirect.com/science/article/pii/S235217911630151X.

[13] R. Schneider, X. Bonnin, K. Borrass, D.P. Coster, H. Kastelewicz, D. Reiter, V.A. Rozhansky, B.J. Braams, Plasma Edge Physics with B2-Eirene, Contrib. Plasma Phys. 46 (1–2) (2006) 3–191, http://dx.doi.org/10.1002/ctpp.200610001, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/ctpp.200610001. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/ctpp.200610001.

[14] D.P. Coster, Reduced Physics Models in SOLPS for Reactor Scoping Studies, Contrib. Plasma Phys. 56 (6–8) (2016) 790–795, http://dx.doi.org/10.1002/ctpp.201610035, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/ctpp.201610035.

[15] I.M. Sobol', On the distribution of points in a cube and the approximate evaluation of integrals, USSR Comput. Math. Math. Phys. 7 (4) (1967) 86–112, http://dx.doi.org/10.1016/0041-5553(67)90144-9, URL https://www.sciencedirect.com/science/article/pii/0041555367901449.

[16] S.E. Davis, S. Cremaschi, M.R. Eden, Efficient Surrogate Model Development: Impact of Sample Size and Underlying Model Dimensions, in: M.R. Eden, M.G. Ierapetritou, G.P. Towler (Eds.), Computer Aided Chemical Engineering, in: 13 International Symposium on Process Systems Engineering (PSE 2018), 44, Elsevier, 2018, pp. 979–984, http://dx.doi.org/10.1016/B978-0-444-64241-7.50158-0, URL https://www.sciencedirect.com/science/article/pii/B9780444642417501580.

[17] B. Ramsundar, R.B. Zadeh, TensorFlow for Deep Learning: From Linear Regression To Reinforcement Learning, First edition, O'Reilly Media, Beijing, 2018, OCLC: on1030582228.

[18] J.H. Friedman, Greedy function approximation: A gradient boosting machine., Ann. Statist. 29 (5) (2001) 1189–1232, http://dx.doi.org/10.1214/aos/1013203451, URL https://projecteuclid.org/journals/annals-of-statistics/volume-29/issue-5/Greedy-function-approximation-A-gradient-boosting-machine/10.1214/aos/1013203451.full. Publisher: Institute of Mathematical Statistics.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine Learning in Python, J. Mach. Learn. Res. 12 (85) (2011) 2825–2830, URL http://jmlr.org/papers/v12/pedregosa11a.html.

[20] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, 2017, ArXiv:1412.6980 [Cs] URL http://arxiv.org/abs/1412.6980.

[21] Jülich Supercomputing Centre, JURECA: Data Centric and Booster Modules implementing the Modular Supercomputing Architecture at Jülich Supercomputing Centre, J. Large-Scale Res. Facil. JLSRF 7 (2021) 182, http://dx.doi.org/10.17815/jlsrf-7-182, URL https://jlsrf.org/index.php/lsf/article/view/182. Number: 0.

[22] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J.F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T.E. Oliphant, Array programming with NumPy, Nature 585 (7825) (2020) 357–362, http://dx.doi.org/10.1038/s41586-020-2649-2, URL https://www.nature.com/articles/s41586-020-2649-2. Publisher: Nature Publishing Group.

[23] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, I. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, S... Contributors, SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python, 2019, ArXiv:1907.10121 [Physics]. URL http://arxiv.org/abs/1907.10121.

[24] W. McKinney, Data Structures for Statistical Computing in Python, Austin, Texas, 2010, pp. 56–61, http://dx.doi.org/10.25080/Majora-92bf1922-00a, URL https://conference.scipy.org/proceedings/scipy2010/mckinney.html.

[25] The pandas development team, Pandas-dev/pandas: Pandas 1.1.3, 2020, http://dx.doi.org/10.5281/ZENODO.4067057, URL https://zenodo.org/record/4067057.

[26] J.D. Hunter, Matplotlib: A 2D Graphics Environment, Comput. Sci. Eng. 9 (3) (2007) 90–95, http://dx.doi.org/10.1109/MCSE.2007.55.

[27] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, 2015, URL https://www.tensorflow.org/.

[28] D. Coster, Characterization of oscillations observed in reduced physics SOLPS simulations, Contrib. Plasma Phys. 58 (6–8) (2018) 666–674, http://dx.doi.org/10.1002/ctpp.201700159, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/ctpp.201700159.

[29] A.S. Kukushkin, S.I. Krasheninnikov, Bifurcations and oscillations in divertor plasma, Plasma Phys. Control. Fusion 61 (7) (2019) 074001, http://dx.doi.org/10.1088/1361-6587/ab1bba, Publisher: IOP Publishing.

[30] R.D. Smirnov, A.S. Kukushkin, S.I. Krasheninnikov, A.Y. Pigarov, T.D. Rognlien, Impurity-induced divertor plasma oscillations, Phys. Plasmas 23 (1) (2016) 012503, http://dx.doi.org/10.1063/1.4939539, URL https://aip.scitation.org/doi/abs/10.1063/1.4939539. Publisher: AIP Publishing LLCAIP Publishing.

[31] C.S. Pitcher, P.C. Stangeby, Experimental divertor physics, Plasma Phys. Control. Fusion 39 (6) (1997) 779–930, http://dx.doi.org/10.1088/0741-3335/39/6/001, Publisher: IOP Publishing.