WPS1-PR(18) 20985

M Blatzheim et al.

Neural Network Performance Enhancement for Limited Nuclear Fusion Experiment Observations Supported by Simulations

Preprint of Paper to be submitted for publication in Nuclear Fusion

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Neural Network Performance Enhancement for Limited Nuclear Fusion Experiment Observations Supported by Simulations

Daniel Böckenhoff[1], Marko Blatzheim[1, 2], Hauke Hölbe[1], Roger Labahn[2], Thomas Sunn Pedersen[1] and the W7-X Team[1]

[1]Max Planck Institute for Plasma Physics, Wendelsteinstraße 1, 17491 Greifswald, Germany
[2]Institute for Mathematics, University of Rostock, Ulmenstraße 69, 18057 Rostock, Germany

daniel.boeckenhoff@ipp.mpg.de, *marko.blatzheim@ipp.mpg.de*

Abstract

It has recently been shown that artificial neural networks (NNs) are able to establish nontrivial connections between the heat fluxes and the magnetic topology at the edge of Wendelstein 7-X (W7-X) [1], a first step in the direction of real-time control of heat fluxes in this device. We report here on progress on improving the performance of these NNs. A particular challenge here is that of generating a suitable training set for the NN. At present, experimental data are sparse, and simulated data, which are much more abundant, do not match the experimental data closely. It is found that the NNs show significantly improved performance on experimental data when experimental and simulated data are combined into a common training set, relative to training performed on only one of the two data sets. It is also found that appropriate pre-processing of the data improves performance. The architecture of the NN is also discussed. Overall a significant improvement in NN performance was seen – the normalized error reduced by more than a factor of three over the previous results. These results are important since heat flux control in a W7-X, as well as in a future fusion power plant, is likely a key issue, and must start with a very limited set of experimental training data, complemented by a larger, but not necessarily fully realistic, set of simulated data.

1 Introduction

In machine learning, the size of the data set on which an artificial neural network (NN) is trained is crucial for a good performance. However, there are many cases where the cardinality of real (i. e. measured) data is limited. This issue may be addressed by complementing the available data with simulations. The underlying model for a simulation is generally not complete. Simplifications are made, in particular to make the model computationally solvable within a reasonable time, given available computing power. Therefore, simulations and experiments often show systematic differences. On the other hand, these reduced models are useful since they reproduce many, and often the most important, measured features of the experimental data.

Other important factors for the performance of a NN are the architecture of the NN [2, 3] and appropriate parametrization [4] of the input data.

These rather general issues are present, and will be addressed in this article, for the specific case of heat loads onto the plasma-facing components of the fusion experiment Wendelstein 7-X (W7-X) [5], which recently went into operation. Experimental data are still scarce, and the interactions between the edge plasma and the plasma-facing components are multifaceted and complicated, involving at least plasma physics, atomic physics, chemistry, and solid state physics, so that state-of-the-art codes such as EMC3-Eirene [6, 7] do not yet capture all the important dynamics. The focus is on developing NNs that, trained on simulated data, supplemented with a very limited number of experimental data, generalize well onto experimental data. Our specific example is the training of a NN using simulated and experimental observations of W7-X heat load patterns to reconstruct ι, a property of the magnetic field at the plasma edge that determines the heat load patterns. The future practical application of this NN will be a part of a real-time control system, ensuring the control and safety of all W7-X plasma facing components.

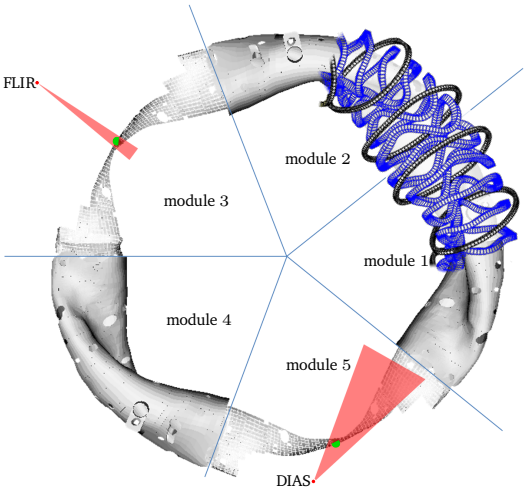
In the next section we briefly explain the essentials regarding W7-X, heat load pattern retrieval from infra-red cameras and a proper formulation of ι, followed by a definition of the data set composition. The two used NN architectures are described and the parametrizations introduced before the NN performance is presented and analyzed.

2 Methods

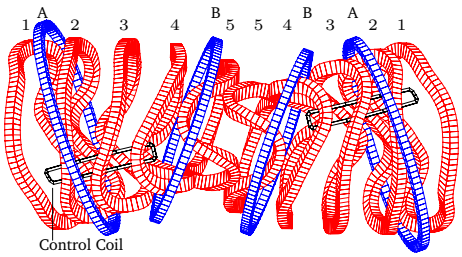
2.1 Context for the Neural Network Application

W7-X is a world-leading, relatively large experiment of the stellarator type [5] (16 m outer diameter, $30\,\mathrm{m}^3$ confinement volume). It has a carefully tailored magnetic field configuration designed to confine hot plasma and aims to explore if this concept can be scaled up to yield net energy production from thermonuclear fusion processes. This paper focuses on the optimization of a neural network that is to become an important piece of a real-time control system for the heat and particles exhausted by the plasma onto the components specially designed to absorb these heat and particle loads. For the present studies, these plasma-intercepting components were so-called limiters (1c), but later divertor modules are used. Perhaps the most important parameter determining the spatial heat load distribution onto these components is the magnetic winding number ι. It describes how many full poloidal turns a magnetic field line performs when performing one full toroidal turn, and is around 0.9 for these studies. Therefore, we started investigating if a NN could determine ι given, as input, measurements by infrared (IR) cameras of heat load patterns. Figure 1a shows an overview of W7-X and IR camera views. Our first results were recently published showing that indeed this is possible [1]. The present paper is a systematic attempt to further improve the performance of NNs for this application, by investigating how the pre-processing of the input data, the character and quantity of training sets, and the NN architecture affect the performance.

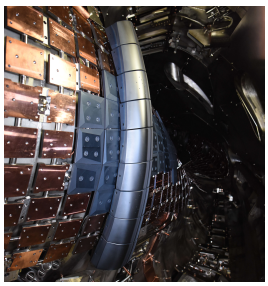
Following the previous paper, we train the NNs to estimate the current in coil set B (Fig 1b), I_B, appropriately normalized, instead of ι itself, but this is a technical detail. For these studies, numerical and experimen-



(a) *Top-down CAD view of the W7-X inner vessel, showing sight lines of the IR camera system for the limiter setup with cutaways in modules three and five as used in the first experimental campaign. On this scale and view, the limiters are small (green). One segment of the total 50 modular (blue) and 20 planar coils (black) is overlaid in modules 1 and 2.*



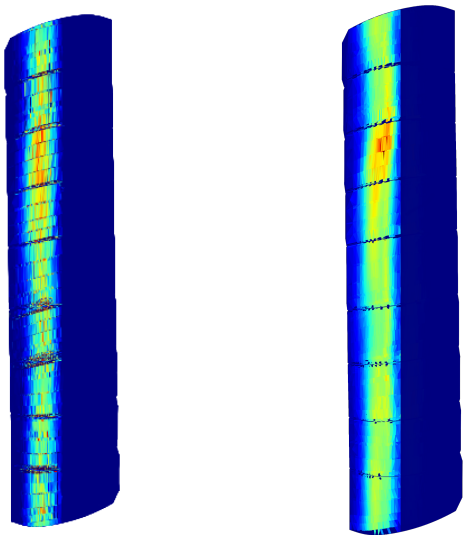
(b) *Coils contained in one module with modular coils $1-5$ (red) and planar coils A, B (blue). One module is point symmetrical towards its center. Adapted after [8]*



(c) *Side view of the W7-X limiter in module 5*

Figure 1: *Overview of essential parts of W7-X in its first experimental campaign*

tal, there is a one-to-one correspondence between the two.



(a) *Heat load, represented by strike point density from a field line diffusion simulation* **(b)** *Heat load, calculated from DIAS camera data*

Figure 2: *Front views of heat load representations on limiter in module* 5 *at discharge "20160309.007". The right side of the limiter is ignored since it is largely obstructed from the camera view.*

For details regarding the underlying physics and data origin (experimental as well as synthetic) we refer to [1]. Figure 2 shows an example of a heat load pattern on the limiter from W7-X module 5 for vacuum field line diffusion simulation (Figure 2a) as well as infrared data (Figure 2b) at the same magnetic configuration. The right half of the limiter is shadowed from the infrared camera view. Some basic characteristics are similar, e. g. the maximum heat load is located at the third limiter tile. However, in detail, the structure of the infrared observation is not reproduced.

2.2 Data Sets

The investigated data sets result from experimental and simulated ι scans of W7-X. The scans were performed by varying the current in one coil set, planar coil set B (cf. [1, Section 2.3]). The simulation set $\mathbb{S}$ was created by the field line diffusion approach described in [1, Section 2.4.2] with $|\mathbb{S}| = 3993$.
The experimental set of processed infrared data $\mathbb{I}$ comes from 16 experiments of the desired ι scan with six different experimental settings(cf. [1, Section 2.4.1]). Each infrared video corresponding to one experiment contributes on average 20 frames for the same value of I_B, leading to a total cardinality of $|\mathbb{I}| = 319$.

A subset $\mathbb{I}_\mathbb{C} \subset \mathbb{I}$ with $|\mathbb{I}_\mathbb{C}| = 190$ is defined such that $\mathbb{I}_\mathbb{C}$ as well as $\mathbb{I} \setminus \mathbb{I}_\mathbb{C}$ cover all six experimental settings. Each of these two sets contain either all or none of the frames of each single experiment.

A mixed set is defined as $\mathbb{M} = (\mathbb{S} \cup \mathbb{I}) \setminus \mathbb{I}_\mathbb{C}$.
To determine NN quality, three disjoint subsets, namely training set, validation set and test set have to be defined.

2.3 NN Architectures

Two NN architectures were compared. Since the data set is very limited, the number of free parameters must be kept low to avoid overfitting. The first one is mainly based on convolutional layers [9] as shown in Figure 3. Three consecutive convolutional layers are followed by two fully connected layers. The second NN architecture starts with an inception module [10] followed by pooling [11], a convolutional layer and two fully connected layers, as shown in Figure 4.
Weights are initialized randomly as recommended in [12] and biases are initialized as zero. The weights and biases are iteratively

improved by the *adam* optimizer [13] during the training process. All activation functions except for the last layer are rectified linear units (ReLU). Because the NNs are designed to solve a regression problem, the activation function of the last layer is the identity. The implementation is done in TensorFlow [14].

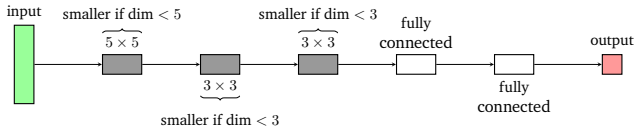


Figure 3: *Structure of NN with three convolutional layers: The grey elements represent convolutional layers with their associated kernel sizes indicated.*

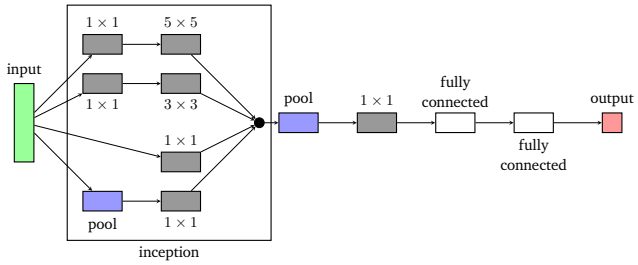


Figure 4: *Structure of NN with inception module: The grey elements represent convolutional layers with associated kernel sizes indicated. The blue elements are max-pooling layers and the white elements are fully connected layer.*

2.4 Parametrization

The heat load is given on an unstructured triangular grid that represents the CAD structure of one limiter. This data is transformed affine from W7-X coordinates to an orthogonal ξ, η, ζ- coordinate system, where ζ points in the direction of $\vec{n}_{\mathrm{m}}$, the normalized mean of the normals of all triangles forming the limiter. The rotation is achieved by the matrix $\boldsymbol{R}$ with

$$\boldsymbol{R} = \begin{pmatrix} 1 - \frac{n_{\mathrm{m},1}^2}{1+n_{\mathrm{m},3}} & -\frac{n_{\mathrm{m},1}n_{\mathrm{m},2}}{1+n_{\mathrm{m},3}} & -n_{\mathrm{m},1} \\ -\frac{n_{\mathrm{m},1}n_{\mathrm{m},2}}{1+n_{\mathrm{m},3}} & 1 - \frac{n_{\mathrm{m},2}^2}{1+n_{\mathrm{m},3}} & -n_{\mathrm{m},2} \\ n_{\mathrm{m},1} & n_{\mathrm{m},2} & n_{\mathrm{m},3} \end{pmatrix}. \tag{1}$$

Each axis is scaled such that the minimal value in this coordinate is 0 and the maximum value is 1. An example for such an affine transformation is shown in Figure 5 for a half cylinder with remote resemblence to the W7-X limiter. Only triangles within a tight bounding box around the limiter are considered for the parametrization.

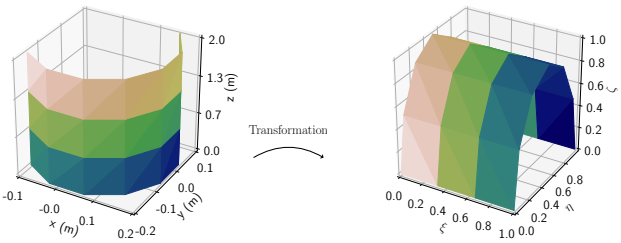


Figure 5: *Sketch of the affine transformation from Cartesian coordinates x, y, z to ξ, η, ζ*

2.4.1 Partitioning

The ξ, η- space is then divided by various 2D grids with sizes described in Table 1 and shown for simulations and experiments in Figure 6a and 6b respectively. There are no divisions in the ζ direction.
The first three columns in Table 1 show one dimensional partitionings. The inception NN is applied only for inputs of dimensionality of at least 5×5. However, for the one-dimensional inputs the convolutional kernel dimensionality is reduced from that presented in Figure 3 such that it does not exceed the input dimensionality.

n_ξ	2	4	9	9	15	18	27	36	72	144
n_η	1	1	1	5	6	8	10	12	15	30
$\mathrm{n}_\xi \cdot \mathrm{n}_\eta$	2	4	9	45	90	144	270	432	1080	4320

Table 1: *Dimensions of the used partitionings and total number of elements*

2.4.2 Extracted Characteristics

For each element p of the partitionings, characteristic values can be extracted. One is the heat-load-weighted spatial mean

$$\vec{\mu}_p = \boldsymbol{\Xi}_p \boldsymbol{W}_p, \tag{2}$$

with triangle centroids

$$\boldsymbol{\Xi}_p = \left((\xi_{pi},\, \eta_{pi},\, \zeta_{pi})_{i=1,\cdots,n_p} \right)^T \in \mathbb{R}^{3\times n_p} \tag{3}$$

and weight vector $\boldsymbol{W}_p$ defined as

$$\boldsymbol{W}_p = \left(\frac{A_{pi} q_{pi}}{\sum_{j=1}^{n_p} A_{pj} q_{pj}} \right)_{i=1,\cdots,n_p} \in \mathbb{R}^{n_p}, \tag{4}$$

with number of triangles per partitioning n_p, triangle areas A_{pi}, and triangle heat load q_{pi} (in Wm^{-2}). The weighted covariance matrix is defined as

$$\boldsymbol{Cov}(\boldsymbol{\Xi}_p) = \left(\boldsymbol{\Xi}_p - \vec{\mu}_p \boldsymbol{J}_{n_p}^T \right) \left(\left(\boldsymbol{\Xi}_p - \vec{\mu}_p \boldsymbol{J}_{n_p}^T \right) \odot \boldsymbol{W}_p \right)^T \in \mathbb{R}^{3\times 3}, \tag{5}$$

with vector of ones $\boldsymbol{J}_{n_p}$ of dimension n_p and element-wise multiplication $\odot$. Another statistical characteristic is the spatial standard deviation $\vec{\sigma}_p$ calculated as

$$\vec{\sigma}_p = \sqrt{\mathrm{diag}\left(\boldsymbol{Cov}(\boldsymbol{\Xi}_p)\right)}, \tag{6}$$

with element-wise square root and the operator diag() which extracts the diagonal elements of a matrix. The direction vector $\vec{\delta}_p$, calculated as the eigenvector corresponding to the largest eigenvalue $\lambda_{\max}$ of $\boldsymbol{Cov}(\boldsymbol{\Xi}_p)$ can be characteristic as well.

This parameter is inspired by divertor heat load patterns which show a more complicated shape [15]. The last examined parameter is the relative heat load

$$\rho_p = \frac{\hat{q}_p}{\sum_{p'=1}^{m} \hat{q}_{p'}} \tag{7}$$

with number of partitions m and

$$\hat{q}_p = \frac{\sum_{i=1}^{n_p} A_{pi} q_{pi}}{\sum_{i=1}^{n_p} A_{pi}} \tag{8}$$

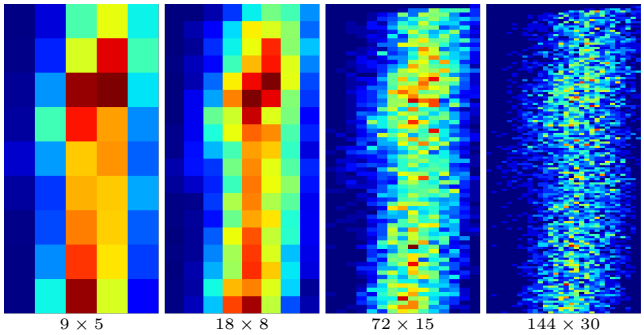
defined as the absolute heat load.
Three combinations of those parameters are studied as NN input: (μ, σ), (μ, δ) and ρ exclusively. In Figure 6 ρ is shown for simulation and experiment of the same physical condition. Note that the input dimension of the NNs is batch size $\times$ n_ξ $\times$ n_η $\times$ input channels, with input channels being 6 for the cases (μ, σ) as well as for (μ, δ) but 1 for ρ.

We define the following notation to describe the NN settings:

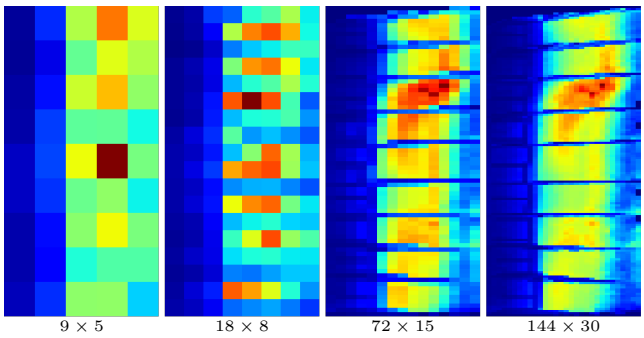
$$\mathrm{f}(\mathrm{train}, \mathrm{validate}, \mathrm{test})^{\text{parametrization}}_{\text{partition, architecture}}, \tag{9}$$

where f can be any function.

For example $rmse(\mathbb{S}_{90}, \mathbb{S}_{10}, \mathbb{I})^{\rho}_{9\times 5, inception}$ refers to the root mean square error of an inception NN trained as well as validated on samples from set $\mathbb{S}$ and tested with experimentally observed infrared data $\mathbb{I}$, requiring a 9×5 input of relative intensities ρ. $\mathbb{S}_{90} \subset \mathbb{S}$ and $\mathbb{S}_{10} \subset \mathbb{S}$ with $\mathbb{S}_{90} \cap \mathbb{S}_{10} = \emptyset$ refer to two disjunct sets for training and validation consisting of $90\,\%$ and $10\,\%$ of the samples within $\mathbb{S}$ respectively. Given a single term in the brackets implies that training, validation and test data sets are disjunct and subsets of the same super set. If one describing parameter is omitted, the entirety of all possible parameters of that kind is referred to. So $(\mathbb{S})^{\rho}_{9\mathrm{x}5}$ describes all NNs parametrized by ρ and partitioned into 9 times 5 parts with both of the two considered architectures. They are trained, validated and tested on subsets of $\mathbb{S}$.



(a) *Simulation*

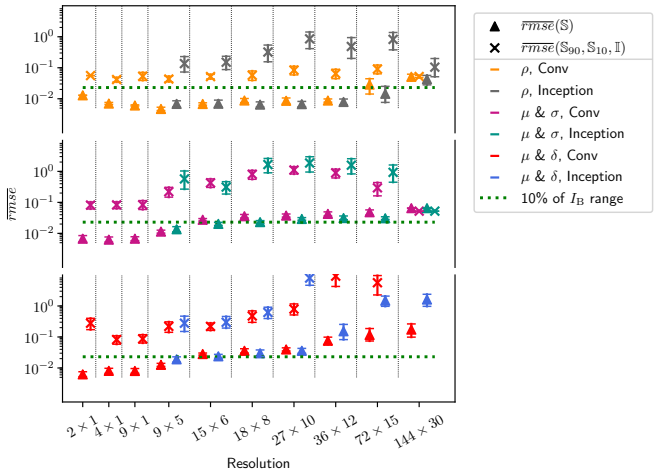


(b) *Experiment*

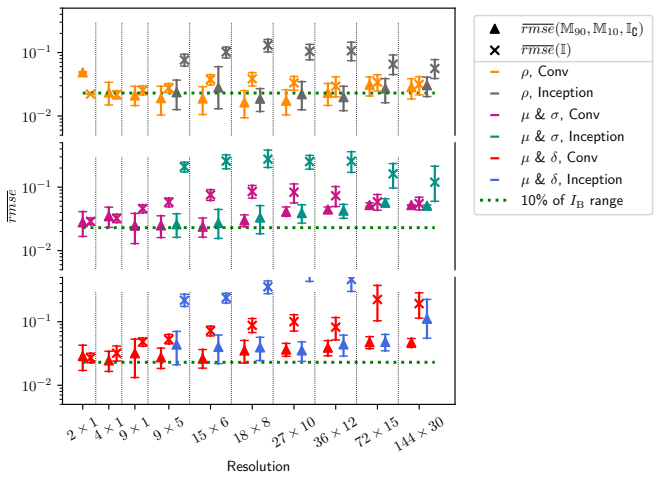
Figure 6: *Comparison of relative heat load between simulation and experiment for four exemplary partitions. The color scale ranges from 0 to the maximal value of each plot respectively.*

3 Results

The NN performance for all tested settings is shown in Figure 7. These two graphics depict the $rmse$ dependence on the partitionings defined in Section 2.4.1. The NN performances measured in terms of the $rmse$ are divided into 10 groups representing the partitioning. Within each group the parameter and NN architecture choices are shown. To avoid false conclusions by statistical outliers, an ensemble of 27 NNs trained with the same settings has been calculated. Variations are the randomness of the weight initialization and the mini batch sampling as well as different learning rates and batch sizes. Markers and bars indicate the mean $rmse$ ($\overline{rmse}$) and the associated $95\,\%$ confidence interval for the mean respectively. The confidence interval has been calculated by bootstrapping [16]. The true possible values of I_B range between -0.05 and 0.18. In order to facilitate the evaluation of the NN reconstruction quality, a reference value of $10\,\%$ of the total I_B range is marked by the dotted, green line. The figures 8, 9, 10, and 12 depict subsets of the outcomes shown in Figure 7 to clarify the observations.



(a) *Logarithmic representation of* $\overline{rmse}(\mathbb{S})$ *and* $\overline{rmse}(\mathbb{S}_{90}, \mathbb{S}_{10}, \mathbb{I})$



(b) *Logarithmic representation of* $\overline{rmse}(\mathbb{M}_{90}, \mathbb{M}_{10}, \mathbb{I}_\mathbb{C})$ *and* $\overline{rmse}(\mathbb{I})$

Figure 7: *Average* $rmse$ *with* 95% *confidence level for various parametrizations, limiter partitionings and NN architectures. In case of smaller* $rmse$ *than* $10\,\%$ *of the* I_B *range a sufficient reconstruction of* I_B *has been achieved.*

3.1 Simulation trained NNs

The results of the NNs trained on $\mathbb{S}$ are shown in Figure 7a. It can be seen that $(\mathbb{S})^\rho$ performs well with both NN architectures for partitionings up to 36×12. The $\overline{rmse}$ is minimal at the partitioning 9×5 and increases with finer as well as coarser resolution. The NNs $(\mathbb{S})^{\mu\&\sigma}$ and the $(\mathbb{S})^{\mu\&\delta}$ perform especially well for the coarse partitionings between 2×1 and 9×5. For finer partitionings the performance decreases gradually. This can be understood contemplating the decreasing information content per section with shrinking section size.

On the basis of Figure 7a we investigate the performance of NNs characterized by $(\mathbb{S})$

vs $(\mathbb{S}_{90}, \mathbb{S}_{10}, \mathbb{I})$. A good performance of the NN $(\mathbb{S}_{90}, \mathbb{S}_{10}, \mathbb{I})$ would be advantageous since it would indicate applicability to new, never conducted experiments. Although it would be preferable, the evaluation indicates that this is not the case.

We observe not only that $\overline{rmse}(\mathbb{S}) \ll \overline{rmse}(\mathbb{S}, \mathbb{S}, \mathbb{I})$, but also that $\overline{rmse}(\mathbb{S}, \mathbb{S}, \mathbb{I})$ significantly exceeds the $10\,\%$ I_B range. The NNs $(\mathbb{S}_{90}, \mathbb{S}_{10}, \mathbb{I})$ are specialized onto patterns of $\mathbb{S}$. Those patterns are not suitable to determine I_B from experimental data. The fundamentally different magnitude of the width of the $\overline{rmse}$ confidence intervals $CI(\mathbb{S}, \mathbb{S}, \mathbb{I})$ as compared to $CI(\mathbb{S})$ points towards the same reason.

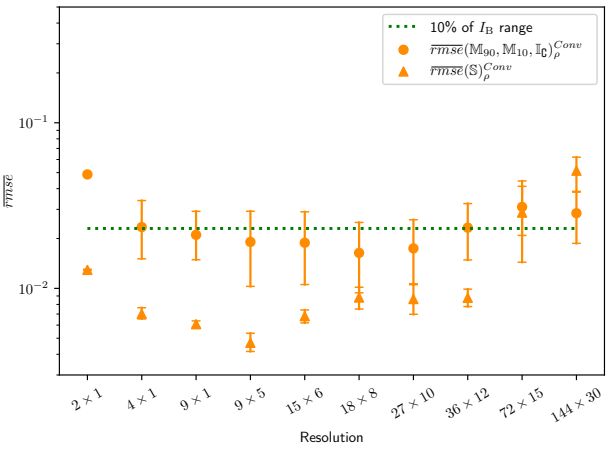


Figure 8: *The development of $\overline{rmse}(\mathbb{S})^{Conv}_{\rho}$ and $\overline{rmse}(\mathbb{M}_{90}, \mathbb{M}_{10}, \mathbb{I}_\mathbb{C})^{Conv}_{\rho}$ are compared for the increasingly refined partitionings. Especially for coarse and extremly fine partitionings the behavior is similiar.*

Both parametrizations including μ show a decreasing performance with growing resolution. Only a marginal difference between $rmse(\mathbb{S})^{\mu\&\delta}$ and $rmse(\mathbb{S})^{\mu\&\sigma}$ can be observed. The $rmse$ $(\mathbb{S})$ range seems independent of the architecture.

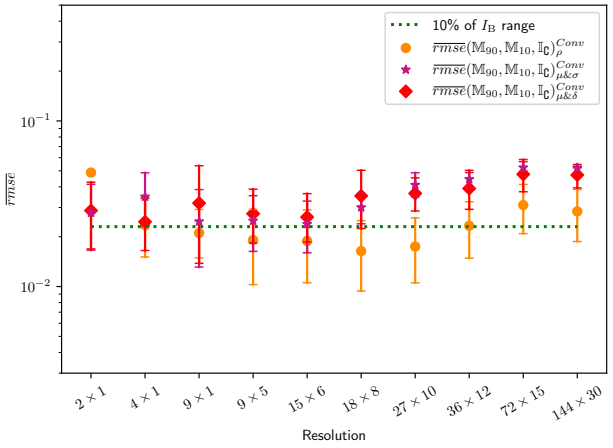


Figure 9: *The development of $\overline{rmse}(\mathbb{M}_{90}, \mathbb{M}_{10}, \mathbb{I}_\mathbb{C})^{Conv}_{\rho}$, $\overline{rmse}(\mathbb{M}_{90}, \mathbb{M}_{10}, \mathbb{I}_\mathbb{C})^{Conv}_{\mu\&\sigma}$ and $\overline{rmse}(\mathbb{M}_{90}, \mathbb{M}_{10}, \mathbb{I}_\mathbb{C})^{Conv}_{\mu\&\delta}$ are compared for the increasingly refined partitionings. Over all partitionings μ & σ and μ & δ behave similary while ρ performes clearly better for fine partitionings.*

3.2 NN trained with simulation and experiment

Since $\mathbb{S}$ based training did not lead to sufficient NN performance for application to $\mathbb{I}$, some samples from $\mathbb{I}$ are provided during training and validation, i. e., $\mathbb{M}$ and $\mathbb{I}_\mathbb{C}$ as defined in Section 2.2 are used. With this procedure, we intend to force the NNs to consider patterns present in both $\mathbb{S}$ and $\mathbb{I}$ during training. The performance of the NNs $(\mathbb{M}_{90}, \mathbb{M}_{10}, \mathbb{I}_\mathbb{C})$ is compared to NNs trained, validated and tested with the small amount of available experimental data only, i. e., $(\mathbb{I}_{37}, \mathbb{I}_4, \mathbb{I}_\mathbb{C})$. Note, that the performance is tested with the same set $\mathbb{I}_\mathbb{C}$. Figure 7b depicts this comparison. The upper end of the occurring $\overline{rmse}$ range is reduced by two orders of magnitude in this Figure as compared to 7a. As in the case of NNs trained with $\mathbb{S}$, the parametrizations by ρ yield the best $\overline{rmse}$ for partitionings between 9×5 and 36×12 while the μ based parametrizations are best

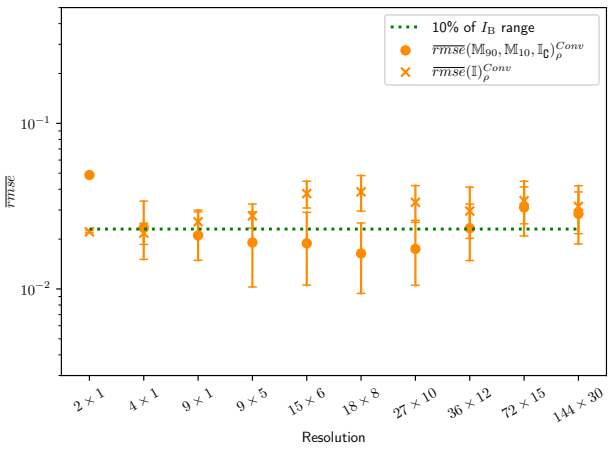


Figure 10: *The development of* $\overline{rmse}(\mathbb{M}_{90},\mathbb{M}_{10},\mathbb{I}_C)_\rho^{Conv}$ *and* $\overline{rmse}(\mathbb{I})_\rho^{Conv}$ *are compared for the increasingly refined partitionings. Especially for partitionings between* 9×5 *and* 27×10 *the mixed training leads to a clearly better NN. The performance for* 18×8 *is shown in more detail in Figure 11.*

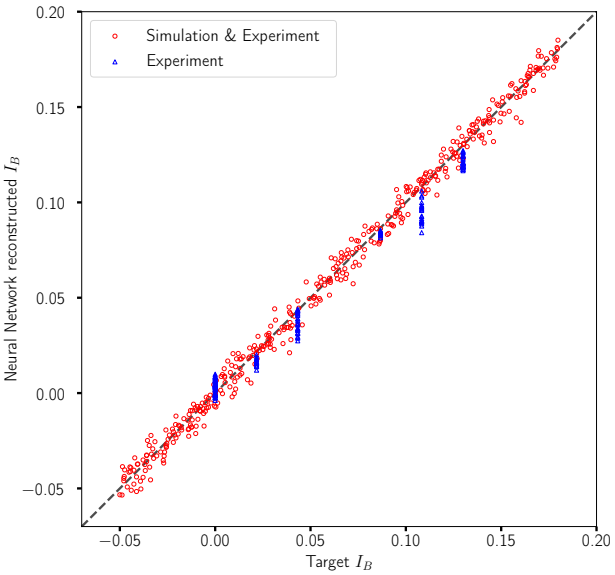


Figure 11: *For the median of* $rmse(\mathbb{M}_{90},\mathbb{M}_{10},\mathbb{I}_C)_\rho^{18\times8,Conv}$ *the fitting performance on the validation set (red circles) and test set* $\mathbb{I}_C$ *(blue triangles) is shown. On the x-axis are the target values and on the y-axis are the NN reconstructions. An ideal result would be the identity, shown by the dashed line.*

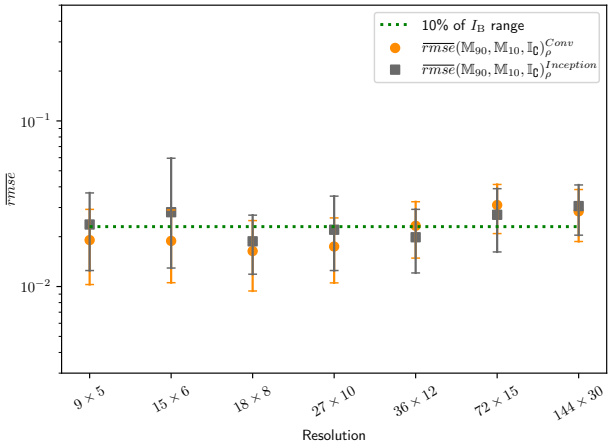


Figure 12: *The performance between the convolutional NN and the inception NN is compared by examining the* $\overline{rmse}(\mathbb{M}_{90},\mathbb{M}_{10},\mathbb{I}_C)_\rho^{15\times6,Conv}$ *with* $\overline{rmse}(\mathbb{M}_{90},\mathbb{M}_{10},\mathbb{I}_C)_\rho^{15\times6,inception}$*. It can not be observed that one architecture is significantly better.*

for one dimensional partitionings decreasing in performance with growing resolution as visualized in Figure 9.
For two-dimensional partitionings, we observe in Figure 10 $\overline{rmse}(\mathbb{M}_{90},\mathbb{M}_{10},\mathbb{I}_C) < \overline{rmse}(\mathbb{I})$, as intended by training on $\mathbb{M}$. In the case of coarse partitioning, training with $\mathbb{M}$ has no advantages above direct training with $\mathbb{I}$.

The most impressive results were found for resolutions between 9×5 and 36×12, for $(\mathbb{M}_{90},\mathbb{M}_{10},\mathbb{I}_C)_{\rho\text{conv}}^{\rho}$. In order to display the good reconstruction quality, we choose a representative $rmse$, namely median $rmse$, of the NNs $(\mathbb{M}_{90},\mathbb{M}_{10},\mathbb{I}_C)_\rho^{18\times8,\text{Conv}}$, as seen in Figure 11. The $rmse$ is 0.008 which is $3.5\,\%$ of the I_B range, so the median $rmse$ is clearly below the $10\,\%$ I_B range threshold. Changing the NN architecture to an inception model seems to slightly but not significantly improve the performance for finer partitioning as seen in Figure 12.

The results clearly outperform those of the NN in [1], where the best $rmse$ was 0.029.

4 Conclusion and Future Work

It was shown here that it is possible to reconstruct an important property of the W7-X edge magnetic field structure from limiter heat load patterns with even better accuracy than earlier reported in [1]. The main challenge was to deal with sparse experimental data given. A naive approach to apply NNs trained and validated with synthetic data to experimental data showed good performance only in a minority of cases. After such a training process most NNs focus on patterns not present in the experimental observations.
For a more targeted training and validation, a mixture of experimental and synthetic data is formed for the training process. This approach resulted in convincing NN performance for certain NN input processing. Partitioning the limiter with resolutions between 9×5 and 36×12 and defining the NN input as the heat load in each part divided by the maximum heat load of all parts results in better performance compared to NNs trained, validated and tested with experimental data only. The low number of experimental results probably leads to overfitting in these nets but the added simulation data diminished these effects. We created NNs that extract relevant patterns from experimental as well as from synthetic data sets to reconstruct an important parameter of the magnetic field at the edge. With this systematic approach NNs were found to outperform the results found in [1].

The upgraded W7-X with installed divertors will be the next object of interest. We will start the investigation with a parametrization based on a two-dimensional partitioning of the heat load. Favoring one of the two examined NN architectures a priori and excluding the other is not possible at this stage because neither consistently outperforms the other. The reached results are satisfactory, however it remains future work to investigate other methods such as generative adversarial nets [17] to further enhance the reconstruction performance when dealing with simulated and experimental data.

5 Acknowledgments

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.
We wish to acknowledge the helpful discussions with G. Leifert, T. Grüning, T. Strauss, M. Weidemann and J. Michael. Our gratitude goes to H. Niemann and F. Pisano for providing the IR data geometry and heat flux mapping. We are also thankful for the support from the CITlab (University of Rostock) and Planet-AI teams.

Bibliography

[1] D. Böckenhoff et al. “Reconstruction of Magnetic Configurations in W7-X using Artifical Neural Networks”. In: *Nuclear Fusion* (2018).

[2] G E Hinton et al. “Improving neural networks by preventing co-adaptation of feature detectors”. In: *arXiv preprint* arXiv (2012), pp. 1–18. arXiv: arXiv:1207.0580v1.

[3] K He et al. “Deep residual learning for image recognition”. In: *IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778. ISBN: 978-1-4673-8851-1. arXiv: 1512.03385.

[4] Dan Cireşan et al. “Multi-column deep neural network for traffic sign classification”. In: *Neural Networks* 32 (2012), pp. 333–338. ISSN: 08936080. arXiv: arXiv:1202.2745v1.

[5] G. Grieger and I. Milch. “Das Fusionsexperiment Wendelstein 7-X”. In: *Physik Journal* 49.11 (1993), pp. 1001–1005. ISSN: 00319279.

[6] Yuhe Feng et al. “EMC3/EIRENE Transport Modelling of the Island Divertor in W7-X”. In: *35th European Physical Society Conference on Plasma Physics. 10th International Workshop on Fast Ignition of Fusion Targets*. Hersonissos, Crete, 2008.

[7] F Effenberg et al. “Numerical investigation of plasma edge transport and limiter heat fluxes in Wendelstein 7-X startup plasmas with EMC3-EIRENE”. In: *Nuclear Fusion* 57.3 (2017), p. 36021.

[8] T. Andreeva. *Vacuum magnetic configurations of Wendelstein 7-X*. Tech. rep. Greifswald: Max-Planck-Institut für Plasmaphysik, 2002.

[9] Yann LeCun and Yoshua Bengio. “Convolutional networks for images, speech, and time series”. In: *The handbook of brain theory and neural networks* 3361 (1995).

[10] Christian Szegedy et al. *Going Deeper With Convolutions*. 2015.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems 25*. 2012, pp. 1097–1105.

[12] Xavier Glorot and Yoshua Bengio. “Understanding the difficulty of training deep feedforward neural networks”. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. 2010, pp. 249–256.

[13] Diederik P. Kingma and Jimmy Lei Ba. “Adam: a Method for Stochastic Optimization”. In: *International Conference on Learning Representations 2015* (2015), pp. 1–15. ISSN: 09252312. arXiv: 1412.6980.

[14] Martín Abadi et al. “TensorFlow: A System for Large-Scale Machine Learning”. In: *OSDI*. 2016, pp. 265–283. ISBN: 978-1-931971-33-1.

[15] Hermann Renner et al. “Physical Aspects And Design of the Wendelstein 7-X Divertor”. In: *Fusion Science and Technology* 46.2 (2004), pp. 318–326.

[16] Thomas J Diciccio and Bradley Efron. “Bootstrap Confidence Intervals”. In: *Statistical Science* 11.3 (1996), pp. 189–228.

[17] Ian J. Goodfellow et al. “Generative Adversarial Networks”. In: (2014). arXiv: 1406.2661.