ISBN: 978-3-95450-248-6 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2025-MOPM079

AUTOMATED CONDITIONING UTILIZING MACHINE-LEARNING: FIRST EXPERIMENTAL RESULTS

S. Wagner*, K. Kümpel, H. Podlech[1], Institute of Applied Physics (IAP), Frankfurt, Germany
[1]also at Helmholtz Research Academy Hesse for Fair (HFHF), GSI Helmholtzzentrum für Schwerionenforschung, Campus Frankfurt, Frankfurt, Germany

Abstract

The conditioning of room temperature cavities is a long process. Additionally, since the cavity or auxiliary equipment can be damaged, constant supervision or extensive safety precautions are required. To reduce the workload for everyone involved and to increase the efficiency of the conditioning process, it was decided to develop a machine learning algorithm with the goal of fully automated conditioning in mind. The initial model was trained on available data of the low energy-domain (up to 500 W). Since it was possible to expand the data to higher power levels during conditionings in 2024, the algorithm is now trained for power levels up to 30 kW. In this paper, the challenges of training with different power scales, as well as the first experimental results shall be discussed.

INTRODUCTION

Conditioning is a necessary but slow process, which every built cavity has to go through at least once. During a conditioning, the power injected into the cavity is slowly increased until the power levels required by the beam dynamics are reached. Since unwanted and potentially dangerous effects, such as outgassing, discharges or multipacting, can occur during this process, the standard procedure at IAP Frankfurt is to conduct a conditioning under the constant supervision of an experienced experimenter.

When a conditioning is carried out by IAP Frankfurt, the forward power P_f, the reflected power P_r and the transmitted power P_t are monitored by the experimenter in addition to the pressure p inside the cavity. Settings of the used signal generator are used to control the conditioning process. Those are the output power, measured in *dBm* and the frequency *f*.

The process itself can take from several days to several weeks, with great differences even between structurally identical cavities. In order to reduce the time and surveillance effort needed to perform a successful conditioning, supporting programs were developed and are further improved. As a first step, a LabView program was designed by IAP, as described in [1]. To achieve the long-term goal of fully automated conditioning, it was decided to develop a trainable machine learning algorithm, as was presented in [2].

* s.wagner@iap.uni-frankfurt.de

DEVELOPMENT PROCESS

Several different algorithms and architectures were tried out during the development process. At the moment, multi-headed deep regression models seem the most promising.

Deep Learning Algorithms

Deep learning algorithms consist of several layers. The first one is the visible input layer. It feeds the input data into the network and has therefore strict restrictions when it comes to its dimensions. After the input layer, several hidden layers are stacked on another. In those, the data is further processed and the weights of the hidden nodes of the layers are adjusted. The last layer is the output layer. It is again visible, and has as an output the desired information.

Training Data

One of the most crucial parts in developing a machine learning algorithm is the preparation of the training data. For the task at hand, a prediction of future values is wanted for the network. For that, a time-skip between input data and the result must be implemented. This time-skip is variable, but for the shown studies it was either 1 s or 2 s, with the scaling working as in Fig.1 shown. Additionally, to reassure a stable learning behavior of the ML-networks, it is important to normalize the data. There are several possibilities for this, depending on what kind of data is used. Since there may be strong outlier values in the data (for example due to sparks), a robust-scaler as described in Eq. 1 was used.

$$\tilde{x} = \frac{x - \bar{x}}{\mathrm{IQR}(x)} \text{ with the inter-quartile range IQR} \qquad (1)$$

Since the algorithm needs scaled data to work, but in the end has to be able to work with vastly different ranges in regards to power, frequency and pressure, some challenges arise.

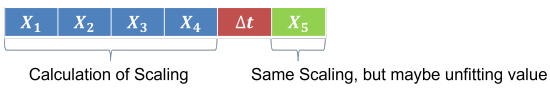

Figure 1: Visualization of the calculation of the scaling for each timestep and the predicted value.

Algorithms

The general structure of the deep learning algorithm used is as described in *Machine Learning Background*. Each

ISBN: 978-3-95450-248-6 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2025-MOPM079

consists of an input layer, several hidden layers, and an output layer.
The available data was divided into training and test data, with roughly 80 % training data and 20 % test data. Of the training data, 20 % were chosen at random as validation splits for each epoch of training.

Convolutional Neural Network Convolutional Neural Networks (CNNs) are most known for their ability to identify images, but they are applicable to every problem in which features can be extracted. During the calculation, a so called kernel goes through the data. It convolutes the data around its center and the so calculated value is further processed, either by another convolutional network or another kind of machine learning algorithm. The adjustable parameters include the number of different kernels as well as their size and what kind of function they apply. Aditionally, the decision has to be made if the dimension of the layer should be preserved by applying so called padding, or if dimensional changes are acceptable.

Recurrent Neural Network Recurrent Neural Networks (RNNs) were developed to work with intrinsically structured data. A prominent example for such problems include time series or natural language. Inside a RNN-cell the data of time-step t_x is concatenated with the results for time-step t_{x-1} and run through an activation function. The output of this cell is then combined with the data of time-step t_{x+1} and put through the same mathematical operations and same weights as the previous data. As a result, the same weights are recurring, resulting in an output that takes also every previous time-step into account. For the here presented work, a more modern version of a recurrent neural network was used, a so called long-short-term-memory (LSTM) RNN. In this version of the network, additional mathematical operations help the network to reinforce (or *remember*) prominent points of the data and to ignore (or *forget*) unimportant points.

Multi-Headed-Regression In an algorithm utilizing a multi-headed structure, the input data is first analyzed by a shared body. Those initial results are then transferred to a number of heads, which are specialized to calculate one desired output. The outputs of all heads is then the over all prediction of the network. The big advantage in this system is, that the heads can be more specialized, while the body extracts information valuable to all desired predictions, as is visualized in Fig.2.

TRAINING RESULTS

For the training, two kinds of depiction were chosen: The comparison between predicted and true values is an obvious approach. This can give a good impression of the overall performance. However, since the scale may vary for the dataset, it is hard to clearly see the performance of the model. A linearized graph is better suited for this. Here, the true value is plotted on the x-axis and the predicted value on the y-axis. For a perfect prediction, a linear graph would

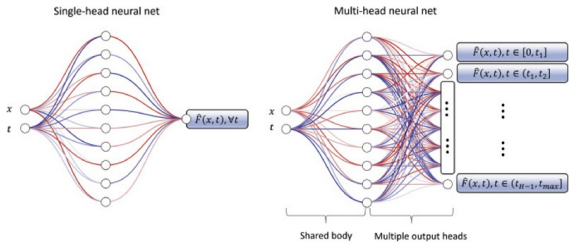

Figure 2: Visualization of a multi-headed-algorithm [3].

follow. As a measurement for each point, how well the prediction was, can the distance between predicted and true value be used. Both graphs are shown in Fig.3. Additionally, the numerical results for different networks are presented in Table 1. The value depicted as baseline error in those graphs corresponds to the scenario, that no change in either frequency or power level was done during the last input time step and the predicted time step.

Table 1: Testing-Errors for different Networks

Type	Input	Avg. Dist.	Avg. Error
LSTM	7 s	0.0382 n.a.	0.0021 n.a.
CNN	6 s	0.0384 n.a.	0.0033 n.a.
CNN-LSTM	11 s	0.3045 n.a.	0.0026 n.a.

EXPERIMENT

Although the algorithm performs well with test data, the most important validation is the experiment. For that, a prototype for the development of a new HLI-RFQ has been prepared for experiments and testing of the code on it.

Experimental Setup

The test setup was deliberately kept simple. Since the amplifier has a maximum output power of 500 W, no cooling is necessary and the danger for the peripheral equipment is kept low. The communication between the used computer and the measuring- and control-equipment is handled via a locally set up LAN-network.
The RFQ in question is a prototype built to validate a new cooling and rod design to suppress mechanical oscillations and increased heating [4].
Since the RFQ was outfitted with transparent windows for laser-vibrometer-measurements, it was well suited to add cameras to observe and record conditioning processes visually. The experimental setup is shown in Fig.4, a visual recording of multipacting in Fig.5.
Additionally, the first experimental results for a fully automated conditioning done by a machine-learning-algorithm is presented in Fig.6. Before the algorithm was given control, an initial power-level was set manually. After that, the algorithm increased the power slowly and reacted to high reflected power-levels, as is shown in the figure. However, after roughly two minutes, it became too ambitious and increased the power-output by several dBm in one second,

ISBN: 978-3-95450-248-6 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2025-MOPM079

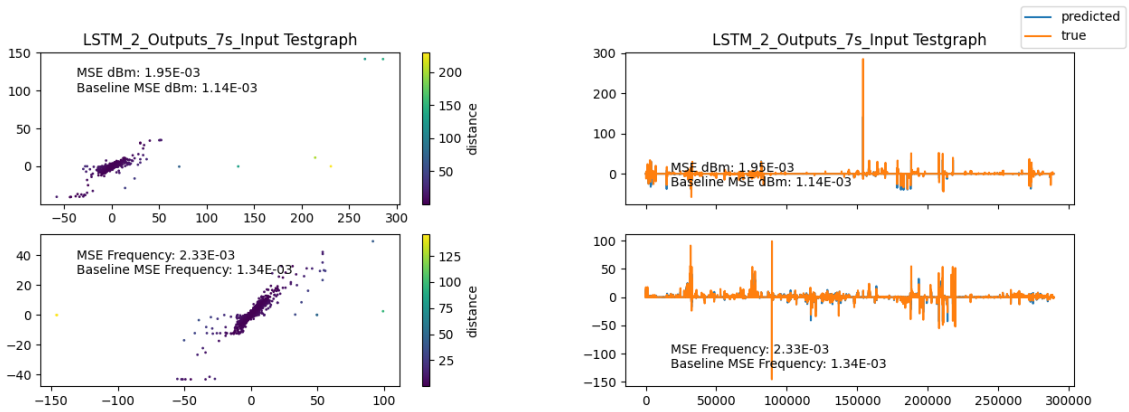

Figure 3: Results for an LSTM-Network for the used test-data, linearized (left) and unscaled predictions (right).

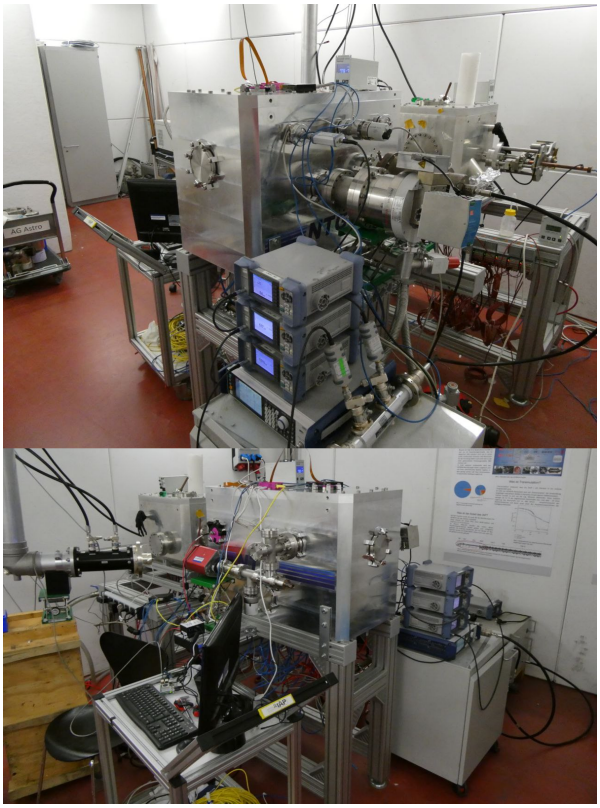

Figure 4: Experimental setup for the validation experiment.

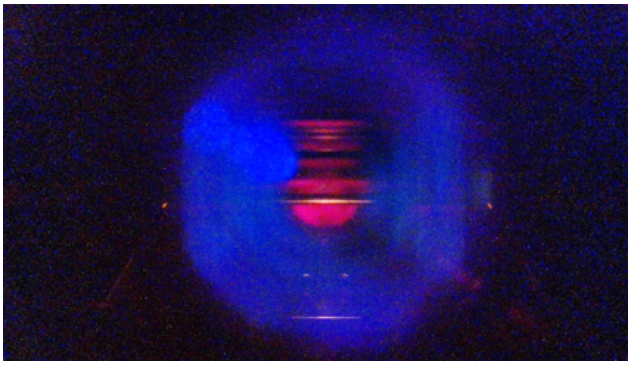

Figure 5: Visual recording of conditioning-effects.

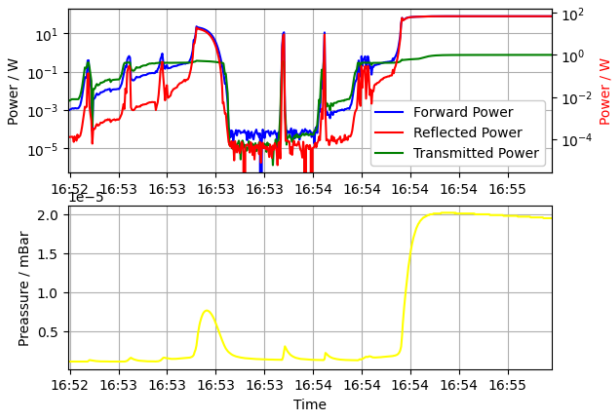

Figure 6: First experimental result for a fully automated conditioning process. The first two minutes of conditioning look promising, but later the network lost control.

resulting in a high reflected power and the ultimate lose of control of the network. After that, the automated process has been interrupted.

SUMMARY AND ROAD-MAP

In this work, the challenges arising from differing input parameter scales for neural networks have been addressed. Furthermore, an improved and updated method for visualizing network performance has been introduced. Results for various multi-headed network architectures—such as CNNs, LSTMs, and combined CNN-LSTMs—have been presented, including the first demonstration of experimental network performance.

While the initial experimental results are encouraging, further improvements are necessary. Planned next steps include a thorough revision of the training data preparation, along with dedicated efforts to optimize and fine-tune the dataset. Additionally, the implementation of a recursive learning approach is under consideration, enabling the network to continue learning during ongoing conditioning processes.

ISBN: 978-3-95450-248-6 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2025-MOPM079

REFERENCES

[1] K. Kümpel *et al.*, "Development and Test of a Program for Automatic Conditioning of Room Temperature Cavities", in *Proc. IPAC'22*, Bangkok, Thailand, Jun. 2022, pp. 2823–2825. doi:10.18429/JACoW-IPAC2022-THPOTK026

[2] S. Wagner, K. Kümpel, H. Podlech, "Automated RF-conditioning utilizing machine learning", in *Proc. IPAC'23*, Venice, Italy, May 2023, pp. 829–831. doi:10.18429/JACoW-IPAC2023-MOPL119

[3] A. Gupta, B. K. Mishra, "Multi-head neural networks for simulating particle breakage dynamics", in *Theoretical and Applied Mechanics Letters*, vol. 14, no. 2, Mar. 2024. doi:10.1016/j.taml.2024.100515

[4] S. Wagner *et al.*, "High Power Tests of a New 4-Rod RFQ with Focus on its Mechanical Vibrations", in *Proc. IPAC'22*, Bankog, Thailand, Jun. 2022, pp. 1523-1525. doi:10.18429/JACoW-IPAC2022-TUPOMS043