INTEGRATION OF MACHINE LEARNING IN HIGH-ENTHALPY PLASMA SPECTROSCOPY

P. E. Hofmeyer, H. Burghaus*, C. Ruhdorfer†, J. Oswald*, A. Bulling†, G. Herdrich**

*Institute of Space Systems, University of Stuttgart, Germany
†Institute for Visualization and Interactive Systems, University of Stuttgart, Germany

ABSTRACT

Optical emission spectroscopy is widely used to characterize high-enthalpy plasmas because it enables measuring a range of plasma parameters. Although the spectra are relatively easy to acquire, extracting meaningful information requires extensive analysis. In this work, a novel approach is developed to automate the analysis of broadband emission spectra by training two machine learning models on synthetic data. The first model is applied to predict plasma temperatures and species number densities in a CO_2 plasma jet. The second model is designed to identify the radiation from potentially occurring species in time-resolved spectra of a titanium material sample demising in an air plasma. Developing a synthetic dataset that allows a trained machine learning model to analyze experimental spectra accurately is identified as a major challenge. Overall, these models offer a significant opportunity to automate the analysis of optical emission spectra.

Index Terms— Machine Learning, Optical Emission Spectroscopy, Temperature prediction, Species detection, Non-equilibrium plasma

1. INTRODUCTION

Plasma wind tunnels are crucial in simulating the extreme conditions encountered during atmospheric entry. New thermal protection system (TPS) material candidates and spacecraft structures are exposed to high-enthalpy flows in these facilities. Experimentally characterizing these plasma flows is an essential task, allowing one to match the conditions during reentry or analyze the demise process of a tested structure [1]. For this, optical emission spectroscopy (OES) is a valuable diagnostic tool that can yield a variety of plasma parameters by measuring and analyzing the light emitted by the plasma itself. The major advantages of OES are the non-invasive nature of the method and the comparably simple experimental setup. However, the interpretation of these acquired spectra can be a complex task, especially in the case of non-equilibrium plasma. [2]

The Institute of Space Systems (IRS) has an expertise in obtaining and analyzing OES data from plasma wind tunnels, which produce complex non-equilibrium plasma flows characterized by different temperatures for different species and degrees of freedom (i.e., excitation, rotation, vibration) [3]. To enable the analysis of large datasets in a reasonable amount of time, a spectral fitting algorithm has been developed at the IRS, reducing the computation time to determine the temperatures and densities of visible plasma species to the order of minutes [4]. However, even with this algorithm, a substantial amount of manual labor and spectroscopic knowledge is required to finetune the algorithm to the visible species in the spectrum.

A promising approach has emerged in recent years for drastically reducing computation time and labor compared to conventional OES data analysis by applying machine learning (ML) to plasma spectroscopy. One of the earliest contributions to this intersection was made by Shadmehr et al. in 1992, who applied ML models to predict process parameters of an etching process from OES data [5]. Since then, with the increase in available computation power and the recent rise in the popularity of Artificial Intelligence, more and more complex use cases have been identified and evaluated at the intersection of machine learning and spectroscopy. For example, Wang et al. [6] demonstrated that these approaches can be used for real-time organic compound detection from OES data while also being interpretable. Or Slimane et al. [6] and Srikar et al. [7] utilized ML to obtain the electron temperature in a plasma from manually pre-processed OES data.

This paper investigates the application of ML to automatically evaluate OES data from plasma wind tunnels without manually selecting spectroscopic features. Here, the reliance on manually labeled data is further reduced by training the models on simulated data with the radiation solver PARADE [9]. This approach is applied to two different tasks. The first task is a regression to determine temperatures and densities of multiple species in a CO_2 plasma, while the second focuses on identifying species in a time-resolved demise test of a titanium sample in an Air plasma.

2. EXPERIMENTAL SETUP AND SOFTWARE

Machine learning, and deep learning especially, require a large amount of high-quality data for good results. However, generating experimental datapoints with plasma wind tunnel facilities at scale is very costly. Furthermore, analyzing the obtained OES data to produce correct labels is a labor-

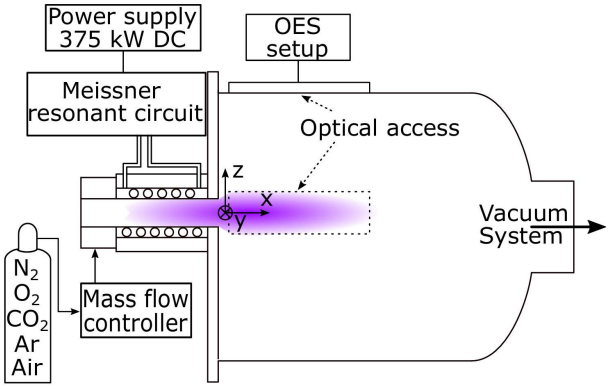

Figure 1: Schematic of the PWK3 facility, including the OES setup

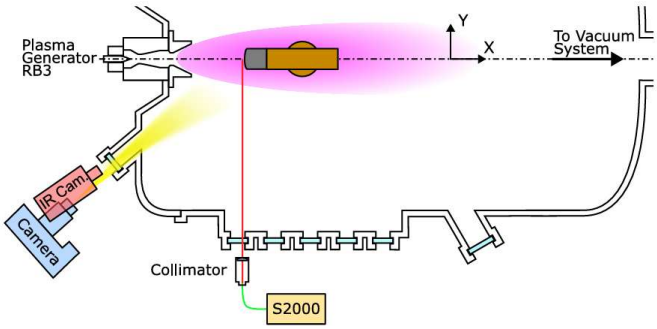

Figure 2: Schematic of the diagnostic setup used for the PRODUCERS campaign at PWK4 (reproduced from [12])

intensive, error-prone process and can carry large uncertainties for the case of determining plasma parameters. Instead, large amounts of high-quality synthetic data for model training are generated in the course of this work. The ML models are then evaluated on experimental data obtained at the IRS.

2.1. Experimental setup

In this work, OES data from experiments in the plasma wind tunnels PWK3 and PWK4 are utilized. PWK3 is comprised of a vacuum chamber and the inductive plasma generator IPG4 that has a maximum anode power of 180 kW. PWK3 is described in detail in [10], [11]. A schematic of the facility, including the position of the spectrometer, is displayed in Fig. 1. PWK4 consists of a vacuum tank and the arcjet plasma generator RB3, which are shown with the optical setup in Fig. 2 [12]. Both facilities are connected to a central vacuum system providing a minimal pressure of 10 Pa [13].

The general operating parameters for the experiments are shown in Table 1. In case of PWK3, the emission of a CO_2 plasma plume is measured with an HR4Pro from Ocean Optics, which records a spectrum between 200 – 1145 nm with a resolution of < 1 nm. The optical setup furthermore consists of a 200 µm fiber from Thorlabs M112L02 and a reflective collimator from Thorlabs RC08SMA-F01.

The data from PWK4 is a time series, recorded at 10 Hz, showing the demise of a titanium sample in an air plasma, recorded during the PRODUCERS [12] campaign. The demise was recorded with the S2000 spectrometer from Ocean Optics, which has an optical resolution of < 1 nm between 180 and 880 nm and was pointed 8 mm ahead of the stagnation point. Furthermore, an in-house produced lens collimator with a focal length of 88 mm, with a spot size of 8 mm at the plasma axis, is used in combination with the 200 µm fiber M092L01 from Thorlabs.

For both spectrometers, the same calibration procedure is performed. Specifically, a Hg(Ar) Pen Ray lamp from Quantum Design is used for the wavelength calibration. The intensity calibration is performed with an Ulbricht sphere (model BN-0102 from Gigahertz-Optik) above 350 nm, while a deuterium lamp (CL3 from Bentham) is used below 350 nm. Because the sensitivity and quality of the intensity calibration method are reduced at the beginning and end of the wavelength ranges, the obtained spectra are trimmed. After cropping, the spectra of the HR4Pro have 2548 pixels between 320 and 970 nm, while the spectra of the S2000 have 1723 pixels between 300 and 877 nm.

Table 1: Operating conditions of the plasma wind tunnel experiments

Parameter	CO2	PRODUCERS
Facility	PWK3	PWK4
Plasma generator	IPG4	RB3
Condition	CO2#01b	Max-H
Working gas	CO_2	Air (N_2/O_2)
Mass flow rate [g/s]	2.2	4
Tank pressure [Pa]	100	27
Electrical power[kW]	160	55.8
Spectrometer	HR4Pro	S2000
Material probe	-	Ti6Al4V

2.2. Plasma radiation database

To create synthetic OES data, precise simulation software is required that can calculate the radiation of atoms and molecules. For this, the Plasma Radiation Database (PARADE), a line-by-line radiation model, for which the development started in 1994 by the IRS and the European Space Agency (ESA), is selected[3]. In this work, the version v3.2 [9] is used with updated atomic emission line information from NIST [14] and the diatomic molecule implementations for AlO and TiO from Loehle et al. [15]. Furthermore, the Python wrapper for PARADE PyWr4P is utilized to produce spectra with multiple excitation

temperatures by simply summing the results of individual PARADE simulations at every wavelength.

2.3. Software tools

Due to the availability of popular ML packages and ease of use, Python [16] is used as the programming language in this work. Definition and training of NNs is done with PyTorch [17] and hyperparameter tuning is performed with Optuna [18]. Data handling and analysis are performed with NumPy [19], Pandas [20], and Scikit-learn [21], while visualizations are created with Matplotlib [22].

3. MACHINE LEARNING METHODOLOGY

Convolutional Neural Networks (CNNs) proved to be a successful architecture for classification tasks at the intersection of spectroscopy and machine learning [6], [23]. Starting from this architecture, a CNN-Residual Network (ResNet) architecture is developed to process OES data to identify species (atoms and molecules) and regress their temperatures and densities in separate tasks. The complete structure, training, optimization, and data processing of our novel architecture are described below, while the results are presented in Section 4.

3.1. CNN-ResNet Architecture

The utilized CNN-ResNet architecture receives a 1-D spectrum as input and calculates its output by passing the input vector through several hidden computation blocks. In our case, the input is first passed through convolutional and then fully connected blocks. Both blocks are visualized in Figure 3.

Our convolutional block consists of a 1-D convolution followed by a layer norm [24] and a ReLU [25] activation function is used as non-linearity. Here, the convolution learns to identify features in the input, while the layer norm speeds up the training process. All convolutional layers have the same properties defined by size, stride, and the number of filters ($Conv_{size}$, $Conv_{stride}$, $Conv_{channel}$). In addition to passing through a convolutional block, the input is also directly passed to the next block via a residual connection, which is empirically known to stabilize training [26]. A MaxPool function is applied to the residual connection to ensure that both branches' output dimensions match.

After a total of N_{conv} layers, the output from the last convolution is passed into a fully connected network. In the case of multiple filters for the convolutional layers, the output is flattened, yielding a 1-dimensional layer that can be input into the first fully connected layer (FCL). The fully connected part of the network consists of a total of N_{FCL} blocks, each comprising an FCL followed by a layer normalization and a ReLU activation function.

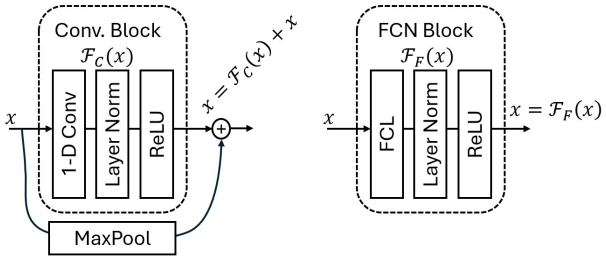

Figure 3: Visualization of a convolutional block (left) and a fully connected block (right). Inspired by [23].

3.2. Regression and classification specifics

Apart from the final layer, the architecture remains consistent for the regression and classification task. In the context of regression, the outputs of this layer correspond to continuous variables, specifically temperatures and densities. Conversely, in classification tasks, the outputs of the final layer indicate the probability that a particular species is present or absent. One can ascertain the visibility status of each species by evaluating these probabilities against a predetermined decision boundary, such as 0.5.

To train the ML model, it is necessary to compute a loss between the model's predictions and the training set's labels. The model is trained to minimize the loss by updating its internal weights using the backpropagation algorithm and gradient descent. For the regression task, the loss is defined with the smooth L1 loss (SL1L) defined as:

$$l_n(x_n, y_n) = \begin{cases} 0.5\dfrac{(x_n - y_n)^2}{\beta}, & if\ |x_n - y_n| < \beta \\ |x_n - y_n| - 0.5 \cdot \beta, & otherwise. \end{cases}$$

Here, x_n is the n-th prediction of the model, while $y_n$is the n-th label of the ground truth, and β is a parameter that controls at which point the loss changes from an L2 to an L1 norm. This loss is less sensitive to outliers compared to the standard L2 norm.

For the classification task, the output of the last layer is first mapped into the range $x \in [0,1]$ with the sigmoid function σ. A loss between the NN output and the ground truth is then calculated via the binary cross-entropy loss:

$$l_n(x_n, y_n) = y_n \cdot \log \sigma(x_n) + (1 - y_n) \cdot \log(1 - \sigma(x_n))$$
$$with\ \sigma(x) = \frac{1}{1 + \exp(-x)}.$$

To obtain a binary prediction from the classifier, the output after the sigmoid function $z_n = \sigma(x_n)$ is compared with a

value called the decision boundary, e. g., 0.5. If z_n is greater than this value, the model predicts that the n-th species is visible in the OES data. With these binary predictions, a performance is calculated for every class via the F1-score. This F1-metric describes the model's accuracy with a score between 0 and 1 and is defined by:

$$F1 = \frac{2TP}{2TP + FP + FN} \quad ,$$

where TP are the true positives, FP the false positives, and FN the false negatives. This metric gives equal importance to the precision, how accurate the identification of a species is, and to the recall, how accurate the prediction that a species is present is.

The coefficient of determination R^2 is utilized to determine the regression model's performance. R^2 can take up values in the interval $(-\infty, 1]$, where 0 means that the model predicts the mean of the observations, and 1 means that the model correctly predicts every test sample. The coefficient of determination is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y_i})^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2},$$

where y_i is the ground truth, $\hat{y_i}$ is the prediction of the model and $\bar{y}$ the mean of the ground truth values.

3.3. Synthetic dataset creation

The synthetic datasets are created with PARADE by randomly varying the available input parameters, i.e., excitation, vibration, and rotational temperatures, as well as the number densities of the radiating species. Specifically, the excitation temperatures are varied between 3,000 – 15,000 K, and the rotational and vibrational temperatures between 1,000 – 20,000 K. The number density spans 7 orders of magnitude for every species and is chosen so that every species is equally likely to appear in the spectrum. This is achieved by preliminary simulations for every species with 9,500 K for the temperatures and setting the density, so that the maximum emission is the same for every species. For carbon, the range is set to 10^{17} - 10^{23} m^{-3}, and all other ranges are set accordingly. The translational temperature is fixed for all simulations to the previously determined value of 3,000 K and is not varied because its effect is not visible in the OES resolution [27]. A total of 100,000 spectra is calculated for every synthetic dataset, and a Sobol sequence samples the random numbers to cover the input parameter space efficiently [28].

The regression dataset for the CO_2 plasma consists of O, C, and CO^+, which were previously identified as the majorly radiating species [4]. In this dataset, 7 parameters

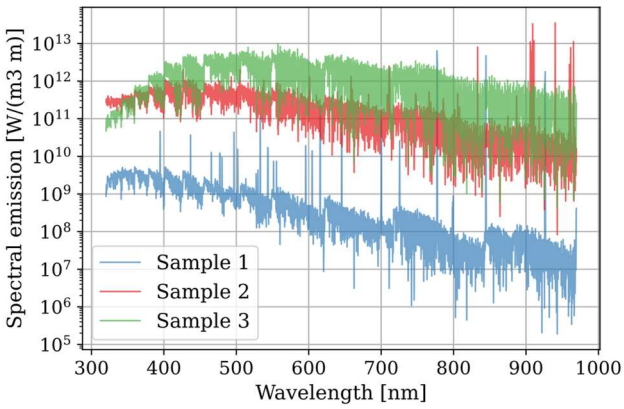

Figure 4: Synthetically generated CO_2 plasma spectra, consisting of O, C and CO^+

are varied: T_{ex}, n_O, $T_{ex,C}$, n_C, T_{rot}, T_{vib}, $n_{CO+,}$ and it is assumed that the excitation temperature of oxygen and CO^+ is equal. Fig. 4 displays three samples of the dataset containing O, C, and CO^+, where the spectral emission is plotted over the wavelength. A logarithmic depiction is necessary to show all features present in these spectra. This is a significant challenge for the machine learning algorithm, as the features are scattered across almost ten orders of magnitude. Furthermore, several features can be hidden, as seen in sample 3, where the emission lines of O and C are masked by the emission of CO^+.

The dataset for the classification task consists of 11 atoms and 3 molecules. In every sample, every species is simulated individually, and the corresponding label is only set to be visible if at least one peak of that species exists with a signal-to-noise ratio of 4 or greater.

3.4. Synthetic data augmentations

Training an ML model on raw synthetic data leads to poor performance when tested on the experimental test set. Specifically, these models make physically unrealistic predictions outside the input parameter range. Therefore, to improve the performance on the experimental data, three different data augmentations are applied to the synthetic dataset.

1. Noise: typical noise values are obtained by subtracting two dark spectra from each other and calculating a standard deviation for the obtained noise. For every pixel of every synthetic spectrum, a random value is calculated using the obtained standard deviation. To mimic the varying sensitivity of the pixel over the wavelength range, the random values are then multiplied by the spectrometer-specific intensity calibration factor curve and applied to the clean synthetic data.

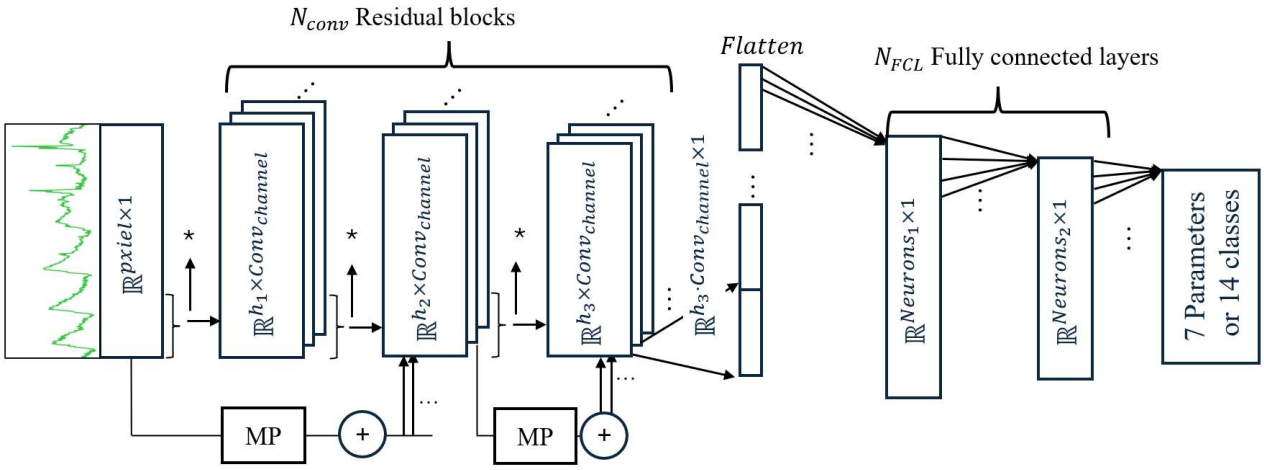

Figure 5: General structure our CNN-ResNet with 3 residual blocks, 2 fully connected layers and a Skip ResNet parameter of 2. The input is a synthetic or measured OES spectrum (on the left in green), while the output is either 7 plasma parameters in the regression task or 14 species in the classification task.

2. Line broadening: Because PARADE does not account for instrument broadening, a Lorentz profile is fitted to the oxygen line at 844 nm, resulting in a full-width half maximum of 0.5 nm. This Lorentz profile is then convoluted with the synthetic spectra.

3. Rolling: A wavelength calibration can never be perfect and is in the range of +/- 1 pixel for the experimental spectra of this work. Thus, the number of samples is tripled by shifting the spectrum one pixel to the left and one to the right.

3.5. Data pipeline

The employed data pipeline processes the model inputs so that all features and labels are between 0 and 1. Because the features, which represent the spectral radiance at a wavelength, span multiple orders of magnitude (see Fig. 4), a logarithmic feature scaling is applied:

$$\tilde{\boldsymbol{x}} = \frac{\log(\max(0, \boldsymbol{x}) + 1) - \log(x_{min}^{train} + 1)}{\log(x_{max}^{train} + 1) - \log(x_{min}^{train} + 1)}.$$

Here, x_{max}^{train} and x_{min}^{train} refer to the maximum and minimum values in the training data, respectively, while x_{min}^{train} is assumed to be at least 0.

In contrast, the labels are scaled according to how they were sampled. That way, the labels have a linear distribution after being scaled. Therefore, every label displaying a density is scaled logarithmically, and every label corresponding to a temperature is scaled linearly.

When applying the ML model to experimental data, the wavelength range of the n-th pixel of the measured data might not coincide with the n-th feature on which the ML model was trained. In this case, the values of the experiment are mapped onto the wavelength ranges of the synthetic data by choosing the closest value available in the measured signal.

3.6. Model training and optimization

The machine learning models are trained on the previously generated synthetic data, separated into a train, validation, and test split of 64%/16%/20% in order to identify and prevent overfitting. A hyperparameter optimization study of 30 trials is performed on the training and validation sets for the classification and regression tasks to find suitable model architectures. Fixed hyperparameters are the maximum number of epochs (200) and the learning rate, which starts at 10^{-4} and is reduced after every 40 epochs of training by a factor of 10. Table 2 displays the varied hyperparameters, the range from which the parameters are sampled, and the values for the optimized ML models for both tasks. The hyperparameters number of neurons, batch size, and β have 5-7 distinct values evenly spread over their available range. After finding the model that achieved the lowest validation score, the model is retrained for as many epochs as the best epoch occurred (in both cases, 200). In Fig. 5 the general structure for the optimized CNN-ResNet in the regression task is displayed. For fair comparison in the regression task, the model score is evaluated with the standard L2 loss function, as their loss function is also varied in the study.

Additionally, the decision boundary is tuned after the final model training for the classification task. Here, the values of the decision boundary are chosen so that the F1-score of every class is maximized on the validation set.

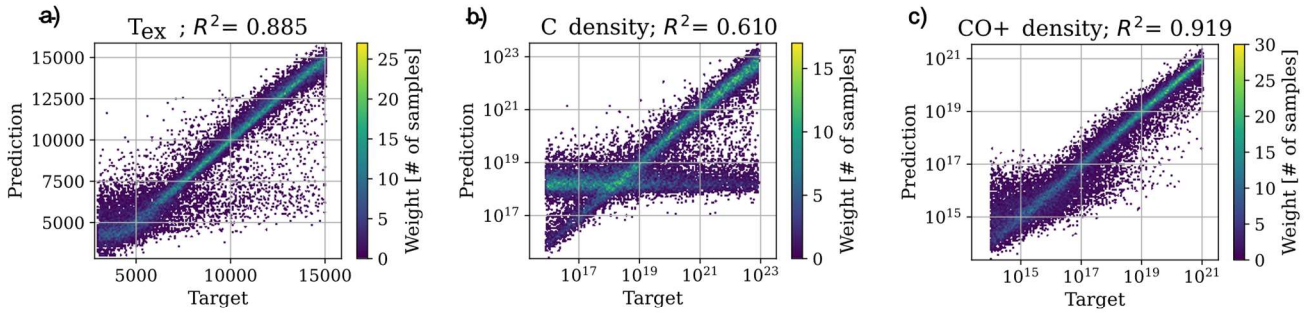

Figure 6: Three exemplary parity plots for the synthetic test set of the regression model. Plot b) and c) are for the labels for which the lowest and highest R2 score are achieved, respectively.

Table 2: Hyperparameter ranges for the optimization and the resulting model architectures for the regression and classification model.

Hyperparameter	Optimization range	Regr.	Class.
N_{conv}	1-8	3	3
Skip ResNet	1,2	2	2
$Conv_{size}$	3, 7, 14	7	3
$Conv_{stride}$	1, 3, 5	5	1
$Conv_{channel}$	1, 4, 8	8	8
N_{FCL}	1 - 5	2	2
Neurons	16 - 2048	256, 16	32, 2048
Batch size	64 - 16384	64	1024
β SL1L	0.1 - 1	0.1	-

4. RESULTS AND DISCUSSION

In this section, the results of the optimized models are presented. For both tasks, regression and classification, the performance of the models is first quantitatively shown on the synthetic data. Then, a qualitative assessment of the models performances on experimental data is given.

4.1. Regression task

The regressor is trained to predict temperatures and number densities of O, C, and CO+ in a CO2 plasma jet. Fig. 6 shows the performance of the ML model on the synthetic test set for 3 of the 7 labels in parity plots, where the predictions are plotted over the ground truth. A perfect model would show a line with a slope of one from the origin. For our model, a concentration of samples is found near that line for every label, highlighted by a brighter color, meaning that more samples are in this area. Additionally, the achieved R^2-scores are given above the plots. Over all 7 labels, the model achieved R^2-scores between 0.61 and 0.92.

In case of the atomic carbon density (Fig. 6.b), two distributions are visible, with the first one being the samples

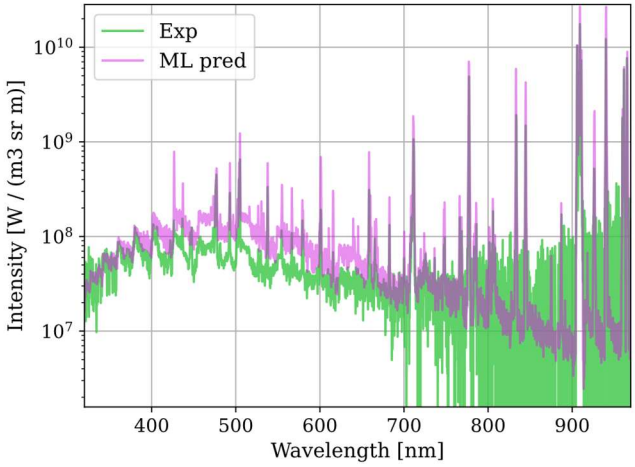

Figure 7: Comparison of a synthetically reconstructed spectrum by the ML prediction with the experimental input spectrum.

that are well predicted on the line through the origin with a slope of one. The second distribution shows a prediction of around 10^{18} regardless of the actual target value. Most points in the second distribution cannot be predicted correctly, because in their respective synthetic spectrum, carbon emission lines are overshadowed, either by the emission of CO^+ or the random noise added to the synthetic data. This also explains why the second distribution contains more samples, where the target has a smaller density, since it is more likely that carbon is not visible in the spectrum. In contrast, the density of CO^+ has the best R^2 score (Fig. 6c), most likely because the molecule's emission is broader and can therefore only be masked by the applied noise, but not atomic emission lines.

The appearance of those two distributions also shows a major challenge when training a machine learning model on labels that are not always visible. Samples that cannot be predicted correctly also introduce uncertainty to samples that should be predictable, which is visible by the outliers at the top right in every parity plot. By utilizing the smooth L1

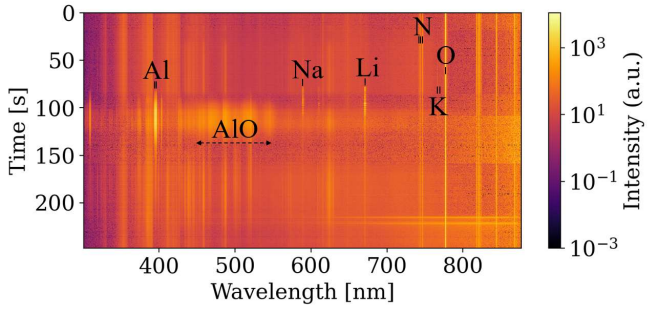

Figure 8: Experimental input data for the classification model, with the emission of some species highlighted (data from [12]).

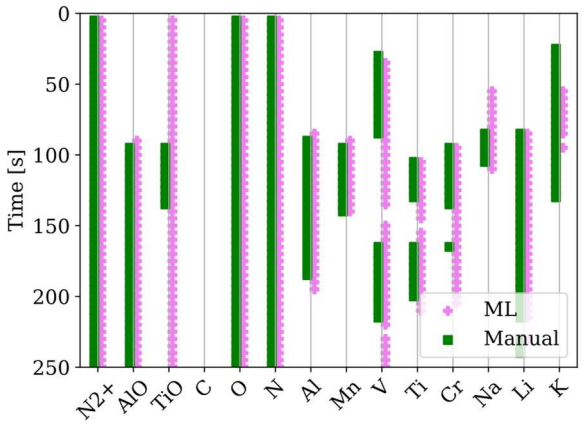

Figure 9: Comparison of the ML model's classifications on the PRODUCERS data with a manual evaluation.

loss, which puts less emphasis on outliers, this problem has been partially addressed, but it remains a challenge.

To show the performance on experimental data, the ML model is applied to a spectrum measured in PWK3 140 mm after the quartz tube of IPG4 and at a radial position of 57mm. This position is chosen as it exhibits atomic and molecular emission. Based on the prediction of the ML model, a spectrum is created with PARADE that can be compared to the original measurement. Fig. 7 shows the comparison of the experimental spectrum and the reconstruction based on the predictions of the ML model. Generally, a good agreement over the whole wavelength range can be observed. While the features of the spectrum are well replicated, a shift in the y-axis persists. Starting from 700 nm, noise is the dominating emission in the experimental data, causing its baseline and that of the ML reconstruction to diverge.

4.2. Classification task

The classifier is trained on the emission of 14 species to predict whether they are visible in a provided spectrum. In the synthetic test set, the model classified all species with similar performance, with F1-scores ranging between 0.75 and 0.85, and an average of 0.80.

Fig. 8 displays the OES data that have been obtained during the PRODUCERS campaign [12] and that is used to assess the model's performance on experimental data. The data consists of time-resolved spectra, documenting the demise of a titanium sample in an air plasma. Additionally, the emissions of some species are labelled in the diagram. The manual evaluation can be ambiguous for species that are only barely visible, as well as the correct identification of AlO and TiO. A more detailed labelling can also be found in [12]. In Fig. 9, the occurrence of the species that the ML model was trained on is plotted over the duration of the experiment for both the manual evaluation and the predictions of the ML model. The predictions for the 2500 spectra are clustered into sets of 50. If in 40 of the 50 spectra, the method predicts that a species is visible, it is shown in the plot. The ML model always correctly identifies the occurrence of the main components of the spectrum, consisting of N2+, O, and N, which is the background radiation of the air plasma. Furthermore, the first appearance of lines of AlO, Al, Mn, V, Ti, Cr, and Li is detected within ±1 evaluation blocks. According to the sampling rate, this refers to a detection within 5s that could be further reduced by clustering fewer spectra together or adjusting the detection threshold in the synthetic dataset. At the same time, the model is less accurate in identifying Na and K, while the worst performance is achieved for TiO. The worse performance for the atomic emission lines is likely due to a small signal-to-noise ratio. In contrast, the detections of TiO emissions could be affected by the radiation of $N2^+$, which is present at similar wavelengths. Finally, the ML model accurately predicts that carbon is not visible in the spectrum, demonstrating potential for models that will be trained on more atomic species than they are observable.

5. CONCLUSION AND OUTLOOK

In this work, an architecture of a convolutional neural network with residual connections (CNN-ResNet) is developed to regress plasma parameters and identify species in emission spectroscopic measurements of high-enthalpy plasma. The model inputs consist of broadband emission spectra, where no prior feature selection is performed. The models are trained on synthetic data generated using the radiation solver PARADAE.

In the regression task, the model is trained on 7 plasma parameters and is applied to the measurement of a CO_2 plasma jet. A significant challenge in this task arises from the potential for certain plasma parameters to be masked in the feature space. This can result in reduced performance, even for samples that could be correctly retrieved. While the general features of the experimental spectrum are correctly

predicted, deploying the system requires further development for more reliable results.

For the species identification, the classifier is trained on the emission spectra of 14 species and applied to the time-resolved measurement of the demise of a titanium sample exposed to a high-enthalpy air plasma flow. Evaluating the model on the experimental dataset indicated that the occurrence times of atomic species can be reliably predicted for prominent lines in the OES data. However, the model displayed problems, especially for TiO, which are likely due to the similar radiation of $N2^+$. Furthermore, the ML models generate results in less than a second, facilitating real-time analysis and processing of large datasets.

Overall, the CNN-ResNet architecture demonstrates a promising approach to automating the analysis of spectroscopic data for high-enthalpy plasma. Future efforts will focus on enhancing the models by examining the training data generation and its augmentation to align synthetic and experimental data distributions better. Additionally, increasing the prediction parameters will broaden plasma conditions where the models can be applied.

Ultimately, a combination of both models is planned, as a symbiosis of the models is expected. Here, the classifier can aid the regressor by determining which species are visible or not, while the regressor can verify the classifier's predictions by providing plasma parameters to produce a synthetic spectrum that can be compared to the original input.

7. ACKNOWLEDGEMENTS

This work was performed on the computational resource bwUniCluster funded by the Ministry of Science, Research and the Arts Baden-Württemberg and the Universities of the State of Baden-Württemberg, Germany, within the framework program bwHPC.

8. REFERENCES

[1] M. Auweter-Kurtz, H. L. Kurtz, and S. Laure, “Plasma generators for re-entry simulation,” *Journal of Propulsion and Power*, vol. 12, no. 6, pp. 1053–1061, Nov. 1996.

[2] U. Fantz, “Basics of plasma spectroscopy,” *Plasma Sources Sci. Technol.*, vol. 15, no. 4, pp. S137–S147, Nov. 2006.

[3] M. W. Winter, “Emissionspektroskopische Untersuchung der Umströmung von Probenkörpern in Hochenthalpen Plasmaströmungen,” Dissertation, University of Stuttgart, 2006.

[4] H. Burghaus, G. Herdrich, and S. Fasoulas, “Derivation of species distribution in inductively heated CO2 plasma via automated spectral fitting,” *Vacuum*, vol. 184, p. 109901, Feb. 2021.

[5] R. Shadmehr et al. “Principal Component Analysis of Optical Emission Spectroscopy and Mass Spectrometry: Application to Reactive Ion Etch Process Parameter Estimation Using Neural Networks,” *J. Electrochem. Soc.*, vol. 139, no. 3, pp. 907–914, Mar. 1992.

[6] C.-Y. Wang, T.-S. Ko, and C.-C. Hsu, “Interpreting convolutional neural network for real-time volatile organic compounds detection and classification using optical emission spectroscopy of plasma,” *Analytica Chimica Acta*, vol. 1179, p. 338822, Sep. 2021.

[7] T. Ben Slimane et al. “Analysis and control of Hall effect thruster using optical emission spectroscopy and artificial neural network,” *Journal of Applied Physics*, vol. 136, no. 15, p. 153302, Oct. 2024.

[8] P. S. N. S. R. Srikar, I. Suresh, and R. K. Gangwar, “Accelerated real-time plasma diagnostics: Integrating argon collisional-radiative model with machine learning methods,” *Spectrochimica Acta Part B: Atomic Spectroscopy*, vol. 215, p. 106909, May 2024.

[9] A. J. Smith et al. “PARADE V3.2 - Final Report,” 9, 2013.

[10] G. Herdrich, M. Auweter-Kurtz, and H. Kurtz, “New Inductively Heated Plasma Source for Reentry Simulations,” *Journal of Thermophysics and Heat Transfer*, vol. 14, no. 2, pp. 244–249, Apr. 2000.

[11] G. Herdrich et al. “Operational Behavior of Inductively Heated Plasma Source IPG3 for Entry Simulations,” *Journal of Thermophysics and Heat Transfer*, vol. 16, no. 3, pp. 440–449, Jul. 2002.

[12] A. S. Pagan, J. W. Oswald, and G. Herdrich, “Experimental Characterization of Spectrographic Responses of Demising Spacecraft Materials,” *submitted to CEAS Space J.*, 2024.

[13] S. Loehle *et al.*, “Assessment of high enthalpy flow conditions for re-entry aerothermodynamics in the plasma wind tunnel facilities at IRS,” *CEAS Space J*, vol. 14, no. 2, pp. 395–406, Nov. 2021.

[14] A. Kramida and Y. Ralchenko, “NIST Atomic Spectra Database, NIST Standard Reference Database 78.” National Institute of Standards and Technology, 1999.

[15] S. Loehle *et al.*, “Extension of the plasma radiation database PARADE for the analysis of meteor spectra,” *Meteorit & Planetary Science*, vol. 56, no. 2, pp. 352–361, Feb. 2021.

[16] G. Van Rossum and F. L. Drake, *Python 3 reference manual*. Scotts Valley, CA: CreateSpace, 2009.

[17] A. Paszke *et al.*, “PyTorch: An imperative style, high-performance deep learning library,” in *Advances in neural information processing systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035.

[18] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A Next-generation Hyperparameter Optimization Framework,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage AK USA: ACM, Jul. 2019, pp. 2623–2631.

[19] C. R. Harris *et al.*, “Array programming with NumPy,” *Nature*, vol. 585, pp. 357–362, 2020.

[20] W. McKinney and others, “Data structures for statistical computing in python,” in *Proceedings of the 9th python in science conference*, Austin, TX, 2010, pp. 51–56.

[21] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in python,” *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[22] J. D. Hunter, “Matplotlib: A 2D graphics environment,” *Computing in science & engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[23] J. Liu et al. “Deep convolutional neural networks for Raman spectrum recognition: a unified solution,” *Analyst*, vol. 142, no. 21, pp. 4067–4074, 2017.

[24] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer Normalization,” Jul. 21, 2016, *arXiv*: arXiv:1607.06450. Accessed: Jun. 20, 2024.

[25] K. Fukushima, “Cognitron: A self-organizing multilayered neural network,” *Biol. Cybernetics*, vol. 20, no. 3–4, pp. 121–136, 1975.

[26] K. He et al. “Deep Residual Learning for Image Recognition,” Dec. 10, 2015, *arXiv*: arXiv:1512.03385. Accessed: Nov. 24, 2024.

[27] H. Burghaus et al. “Aerothermodynamic Characterization of Inductively Generated Carbon Dioxide Plasma by Laser Absorption Spectroscopy,” *Journal of Thermophysics and Heat Transfer*, vol. 38, no. 1, pp. 98–108, Sep. 2023.

[28] I. M. Sobol’, “On the distribution of points in a cube and the approximate evaluation of integrals,” *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 4, pp. 86–112, Jan. 1967.