J Therm Spray Tech
https://doi.org/10.1007/s11666-025-01965-x

ORIGINAL RESEARCH ARTICLE

Physics-Informed Neural Networks for Predicting Particle Properties in Plasma Spraying

K. Bobzin[1] · H. Heinemann[1] · A. Dokhanchi[1]

Submitted: 20 June 2024 / in revised form: 17 January 2025 / Accepted: 6 February 2025

Abstract Plasma spraying is a key industrial coating process that exhibits intricate nonlinear interactions among process parameters. This complexity makes accurate predictions of particle properties, which greatly affect process behavior, very challenging. Specifically, particle velocities and temperatures profoundly impact coating quality and process efficiency. Conventional methods often require empirical correlations and extensive parameter tuning due to their limited ability to capture the underlying physics within this intricate system. This study introduces physics-informed neural networks (PINNs) as a solution. By seamlessly integrating known physical laws and constraints directly into the model architecture, PINNs offer the potential to learn the underlying physics of the system. For comparison, artificial neural networks (ANNs) are also developed. Computational fluid dynamics simulations of a plasma generator and plasma jet model provide data to train both ANN and PINN models. The study reveals an improvement in particle velocity prediction through the proposed PINN model, demonstrating its capability to handle complex relationships. However, challenges arise in predicting particle temperature, warranting further investigation. The developed models can aid in optimizing the plasma spraying process by predicting essential particle properties and guiding necessary process adjustments to enhance coating quality.

Keywords artificial neural networks (ANNs) · atmospheric plasma spraying (APS) · computational fluid dynamics (CFD) · gray box model · machine learning (ML) · physics-informed neural networks (PINNs)

Introduction

Given the intricate interactions among various influencing factors and the challenging conditions involved in plasma spraying, characterized by extremely high temperatures and velocities, the task of optimizing spraying parameters poses a significant challenge. Depending solely on an empirical method for this optimization is not feasible and would consume considerable time. A more effective strategy involves numerical analysis, such as computational fluid dynamics (CFD), to comprehend the intricate relationships within the plasma spraying process. Consequently, extensive research in the literature has focused on modeling momentum, heat, and mass transfer to injected powder particles through CFD. This is exemplified by references such as (Ref 1) for atmospheric plasma spraying (APS) or (Ref 2) for high velocity oxygen fuel (HVOF) process. However, sophisticated CFD models suffer from the drawback of demanding substantial computation time, particularly in the case of complex systems like the multi-arc plasma spraying process (Ref 3). The high

This article is an invited paper selected from presentations at the 2024 International Thermal Spray Conference, held April 29–May 1, 2024, in Milan, Italy, and has been expanded from the original presentation. The issue was organized by Giovanni Bolelli, University of Modena and Reggio Emilia (Lead Editor); Fardad Azarmi, North Dakota State University; Sara Bagherifard, Politecnico di Milano; Partha Pratim Bandyopadhyay, Indian Institute of Technology, Kharagpur; Šárka Houdková, University of West Bohemia; Heli Koivuluoto, Tampere University; Yuk-Chiu Lau, General Electric Power (Retired); Hua Li, Ningbo Institute of Materials Technology and Engineering, CAS; Sinan Müftü, Northeastern University; and Filofteia-Laura Toma, Fraunhofer Institute for Material and Beam Technology.

✉ A. Dokhanchi
ali.dokhanchi@rwth-aachen.de

[1] Surface Engineering Institute (IOT), RWTH Aachen University, Aachen, Germany

Published online: 26 February 2025

computational cost linked to simulating the plasma spraying process poses a barrier to simulation speed and reduces its competitiveness in the real-time industry 4.0.

An efficient solution for swiftly replicating CFD simulations in plasma spraying involves integrating simulation models with artificial intelligence (AI) and machine learning (ML) algorithms, thanks to their relatively low prediction overhead and robust capacity to learn from big data. This approach is expected to result in a trade-off between computational cost and accuracy, balancing the relatively fast ML predictions with the time-consuming nature of simulations. Earlier investigations have applied AI methodologies, predominantly utilizing experimental data, to predict process parameters for achieving desired in-flight particle properties or coating characteristics. For instance, Kanta et al. (Ref 4) utilized artificial neural networks (ANNs) and fuzzy logic (FL) to predict in-flight particle properties based on process parameters for the deposition of alumina-titania using APS. Mojena et al. (Ref 5) created an ANN model to forecast the rate of erosive wear of hard coatings deposited by HVOF and flame spray techniques under various operational conditions. Only a few studies have employed simulation data sets for training ML models in thermal spraying. For example, Zhu et al. (Ref 6) utilized simulation results of in-flight particle characteristics to construct a convolutional neural network (CNN) model. This model was designed to extract potential features of in-flight particle characteristics distributions at each two-dimensional cross section along the spraying direction.

In a prior study (Ref 7), we created efficient and accurate metamodels for forecasting average particle properties at specific stand-off distances in APS. This was achieved by employing two distinct ML methods, in particular support vector machine (SVM) and residual neural network (ResNet), trained with CFD simulations of the plasma jet. The findings indicated that the metamodels developed could predict average particle properties with significantly greater accuracy compared to the properties of individual particles. In a subsequent study (Ref 8), we developed an ML approach with two consecutive ResNets using simulation data sets. This approach enabled the precise prediction of individual particle properties in relation to their positions, facilitating the replication of simulated particle trajectories within the plasma jet. It should be mentioned that ensuring accurate point-wise predictions for every data point could not be guaranteed.

Physics-informed neural networks (PINNs), a relatively recent variant of ANNs, integrate physical laws into the learning process by incorporating the residuals of a system of partial differential equations (PDEs) into the loss function of the neural network (Ref 9). PINNs have demonstrated their performance capability in providing accurate predictions, particularly in the field of fluid mechanics (Ref 10). Furthermore, PINNs provide the advantage of transforming the conventional black box model of ANNs into a more transparent gray box model by incorporating physical laws. Despite the powerful features inherent in PINNs, its application in the field of thermal spraying is notably scarce in the literature. In a recent study, Gui et al. (Ref 11) proposed a hierarchical neural network combining PINNs and CNN to develop prediction models for in-flight particle properties and the performance of NiCr-Cr_3C_2 coatings in HVOF process.

This study aims to bridge the existing gap by leveraging the potent capabilities of PINNs for predicting particle properties in multi-arc plasma spraying. To the best knowledge of the authors, this work represents a pioneering effort in the application of PINNs to APS. The construction of PINN models for predicting in-flight particle velocities and temperatures involves using a subset of the relevant equations that underlie the CFD simulation models. Additionally, ANN models are developed to compare the results from the ANN and PINN models, assessing the potential of PINNs in addressing challenges within the context of plasma spraying. Both ANN and PINN models are developed using simulation data sets.

Methods

This chapter provides an overview of ANNs and PINNs. It describes the data preparation process, details the model setup, and outlines key methodological considerations. Finally, the results are presented and analyzed in the next chapter.

Artificial Neural Networks (ANNs)

ANN is a robust machine learning method that mimics the human brain's interconnected neurons, allowing it to learn complex patterns and make intelligent decisions. ANNs consist of layers of interconnected nodes, or artificial neurons. These nodes process information and adapt their connections based on the encountered data, enabling the network to generalize and perform tasks such as classification, regression, and pattern recognition. ANN has proven its efficacy in various fields, utilizing specialized variants like convolutional neural networks (CNN) (Ref 12) for image recognition and recurrent neural networks (RNN) (Ref 13) for processing time-series data. Figure 1 illustrates a representative structure of a fully connected feed-forward ANN, featuring two inputs, three outputs, and a single hidden layer.

To evaluate the model, a loss function (L) is computed at the network's conclusion. The mean square error (MSE)

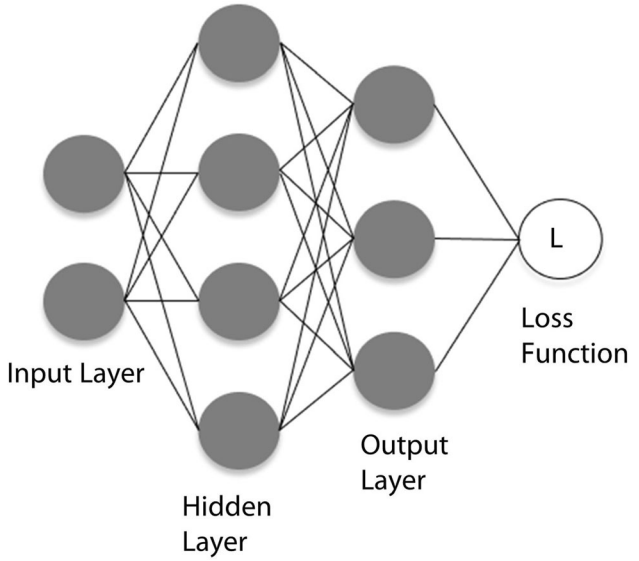

Fig. 1 Exemplary illustration of an ANN structure

stands out as one of the most employed loss functions, representing the mean L2-norm of the difference between the predictions p and the targets t within N sample points, as shown in Eq 1.

$$L_{\mathrm{MSE}} = \frac{1}{N}\sum_{n=1}^{N} \|p - t\|_2 \qquad \text{(Eq 1)}$$

Physics-Informed Neural Networks (PINNs)

PINNs are specifically designed for encoding any underlying law of physics described by general nonlinear PDEs that govern a given data set. This technique utilizes known physical laws during the training process to improve the performance of neural networks in modeling physical systems, especially when only limited data are available (Ref 14). In addition, PINNs can predict important constant values in PDE systems that are not known and need to be determined during the training process. An example PINN structure is shown schematically in Fig. 2. The PINN loss function comprises two essential components: the data-driven loss term and the physics-informed loss term. The data-driven loss term focuses on minimizing the disparity between the predicted values of the neural network and the target data, ensuring a robust fit to the available data sets. Concurrently, the physics-informed loss term integrates the inherent physical principles of the system, expressed by the given equations. This term enforces the network's solutions to comply with the underlying physics of the problem, guiding the model to generate predictions that are not only data-consistent but also adhere to the governing laws of the system.

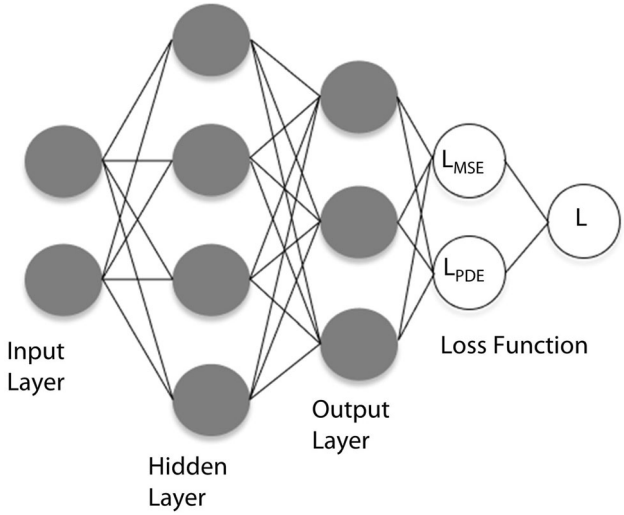

Fig. 2 Exemplary illustration of a PINN structure

The final loss is a weighted combination of the data-driven loss (L_{MSE}) and the physics-informed loss (L_{PDE}) according to Eq 2.

$$L_{\mathrm{PINN}} = L_{\mathrm{MSE}} + \alpha L_{\mathrm{PDE}} \qquad \text{(Eq 2)}$$

In Eq 2, the term α is a hyperparameter that controls the weight assigned to the physics-informed term relative to the data-driven term. Adjusting this hyperparameter allows the model to balance the influence of data fitting and adherence to physics during the training process. The loss function L_{MSE} corresponds to the mean squared error defined in Eq 1. Readers may refer to (Ref 15) for mathematical calculations of the physics-informed loss L_{PDE} across various classical problems in fluid mechanics.

Data Preparation

The simulation data sets used to train the ML models in this study are derived from a previous numerical model simulating the plasma spraying process of a three-cathode plasma generator at the Surface Engineering Institute (IOT) at RWTH Aachen University. This model, implemented using ANSYS CFX version 20.2 (ANSYS, Inc., USA), is partitioned into two sub-models: one for the plasma generator and another for the plasma jet. In the plasma jet model, a two-way coupled approach is adopted, considering plasma-particle interaction to examine the mutual influences of the plasma on the particles and vice versa (Ref 1). A comprehensive description of the numerical modeling utilized in this study is available in (Ref 16) and (Ref 17). To train the ML models, we utilize data from 45 simulations initially configured through a design of experiment (DoE) using latin hypercube sampling (LHS). LHS systematically samples from the input space by dividing each parameter interval into distinct and equally

probable subranges, ensuring comprehensive coverage of the parameter space during sampling. The ranges of process parameters for the DoE method are provided in Table 1. The simulation data used in this study originate from our previously mentioned study in the introduction, which focused on predicting particle properties in plasma spraying using machine learning models SVM and ResNet (Ref 7). As described in that paper, the structure of the simulation data collected using the latin hypercube sampling (LHS) method can be described as follows: Exemplarily, the simulation data sets for the parameters primary gas flow, electric current, carrier gas flow, powder feed rate, particle size distribution and stand-off distance, respectively, are:

1. 40.36 SLPM, 461.6 A, 6.39 SLPM, 28.8 g/min, − 35 + 15 µm, 126 mm
2. 40.36 SLPM, 532.9 A, 5.72 SLPM, 15.6 g/min, − 35 + 15 µm, 153 mm
3. 41.37 SLPM, 473.8 A, 4.04 SLPM, 12.0 g/min, − 35 + 15 µm, 169 mm

⋮

45. 59.87 SLPM, 470.3 A, 4.04 SLPM, 18.0 g/min, − 75 + 55 µm, 144 mm

As seen in the structure above, within each of the 45 simulations, only the particle size can vary in the specified range, while the other five process parameters remain constant. During each simulation, 2000 particles are injected into the plasma jet, and the properties of these particles are extracted at a specific stand-off distance from the substrate. However, due to different process parameters of each simulation, not all the 2000 simulated particle trajectories reach the specified stand-off distance determined by the LHS method for that simulation. Therefore, the exact number of output data per simulation is not the same and can vary between 1500 and 2000 particle trajectories. The inputs and outputs of each simulation are provided with indices to be able to assign the particles of each simulation for the ML models.

Table 1 Configuration of parameters for the DoE method to generate simulation data

Parameter, unit	Interval
Primary gas flow, SLPM	40–60
Electric current, A	400–540
Carrier gas flow, SLPM	3.5–7
Powder feed rate, g/min	10–30
Particle size distribution, µm	− 35 + 15; − 55 + 35; − 75 + 55
Stand-off distance, mm	100–180

The simulations investigate the multi-arc plasma spraying process for ceramic feedstock materials, using alumina as a representative example due to its well-documented material properties in the literature. The specific heat capacity of alumina is modeled as a function of both temperature and physical state, utilizing NASA polynomials (Ref 18). The latent heat of fusion and boiling is determined based on enthalpy values at the respective phase transition temperatures. Additionally, enthalpy calculations are performed using NASA polynomials. To simplify the analysis, a constant thermal conductivity of $\kappa = 6$ W/mK, representative of high-temperature conditions, is assumed for the alumina feedstock. The numerical model considers particle trajectories along with corresponding changes in temperature, velocity, size, and thermophysical state (solid, liquid, or gas).

Model Setup

For each of both the PINN and ANN techniques, two separate models are created; one specifically designed for predicting particle velocity and the other for predicting particle temperature. The construction of these models is executed using MATLAB. The inputs of the ML models are the parameters listed in Table 1. To reduce the volume of the large training data for the models, this study considers sampled data points near the substrate. Furthermore, the exported data encompasses solely the particle and fluid features along the particle trajectories. Hence, a comprehensive analysis of the entire fluid field and the boundary conditions is not conducted during the training in this study.

The hyperparameters for ANN include the learning rate, number of hidden layers, number of neurons in each hidden layer, mini-batch size, and the number of training epochs. The mini-batch size refers to the number of training examples utilized in a single iteration (one update of the model's weights), while the number of training epochs represents the total number of passes through the entire training data set during model training. In this study, following a hyperparameter tuning, a single-layer perceptron with a mini-batch size of 1000 was utilized for ANN models. The architecture comprised 40 neurons for velocity and 30 neurons for temperature.

All input data were normalized within the range of zero and one. This normalization aims to balance input scales during training. It is crucial to note that, as a consequence, the output must be back-transformed into the original scale in each iteration. The physics-informed loss term in PINNs involves the residuals of the equations, and these residuals, when normalized, may not exhibit a linear correspondence

to the original, unscaled values. Back-transforming the normalized outputs into the original scale becomes essential during each iteration to ensure that the computed loss accurately reflects the physical constraints and relationships inherent in the original problem space.

For the particle velocity prediction in the PINN setup, Eq 3 was employed to compute the physics-informed loss term. This equation characterizes the movement of a particle with mass m_{p} in Lagrangian form, specifically by constraining the force equilibrium between the plasma jet and particles to the viscous drag force (Ref 1). In Eq 3, v_p and A_p are defined as the particle velocity and cross-sectional area. Additionally, ρ_g and v_g represent the density and velocity of the plasma gas. The drag coefficient is introduced as C_{D}, and its value depends on the particle Reynolds number. The values and properties of particles and plasma gas were derived from the exported simulation data along the particle trajectories in the plasma jet.

$$m_{\mathrm{p}}\frac{d_{v_p}}{dt} = \frac{1}{2}C_{\mathrm{D}}A_p\rho_g\left|v_p - v_g\right|\left(v_g - v_p\right) \qquad \text{(Eq 3)}$$

The derivative $\frac{d_{v_p}}{dt}$ in Eq 3 can be computed by Eq 4, where y is the particle coordinate in the spray direction. The derivative terms are computed using MATLAB automatic differentiation (AD) in each training iteration. AD is a set of techniques designed to efficiently and accurately evaluate derivatives (gradients) of numeric functions expressed as computer programs (Ref 19).

$$\frac{d_{v_p}}{dt} = \frac{d_{v_p}}{dy}\frac{dy}{dt} \qquad \text{(Eq 4)}$$

Similar to the approach used for velocity prediction, Eq 5 was employed to determine the physics-informed loss term for the prediction of particle temperature. This equation characterizes the rate of change in particle temperature, expressed in the form of the energy equation for a single particle (Ref 1).

$$m_{\mathrm{p}}c_{\mathrm{p}}\frac{d_{T_p}}{dt} + Q_{\mathrm{melt}} - Q_{\mathrm{resolid}} + Q_{\mathrm{vap}} = \pi d_p\kappa_g Nu_p\left(T_g - T_p\right) \qquad \text{(Eq 5)}$$

In Eq 5, c_{p} represents the specific heat capacity of the particle, and Q_{melt}, Q_{resolid}, and Q_{vap} denote the energy involved in the particle melting, resolidification and vaporization process. The particle diameters and thermal conductivity of the plasma gas are expressed as d_p and κ_g. The particle Nusselt number is denoted by Nu_p. Additionally, the gas temperature and particle temperature are represented by T_g and T_p. In this study, for the purpose of simplification in the training process, we neglect the term Q_{vap} and the portion of particles in the gas phase. This simplification is based on the observation that the gas mass fraction remains low relative to the solid mass fraction throughout the simulation. The derivatives for temperature are calculated using AD technique, like the computation of derivatives for velocity according to Eq 4.

Results and Discussion

In the following, we first present and discuss the results of the ANN models. Subsequently, the outcomes of the PINN models are presented, providing a comparison of their respective performances.

Figure 3 shows the results of the mean particle a) velocities and b) temperatures per simulation from the ANN models. The mean values predicted by the ANN models shown in red are denoted with "Mean ANN," while the corresponding target values from the simulation model displayed in blue are labeled with "Mean Sim." The mean values over all 45 simulations are also depicted in this figure and denoted by "grandmean." To evaluate the prediction accuracy of the individual particle properties, the mean absolute percentage error (MAPE) is calculated. This performance metric is calculated to be $\mathrm{MAPE}_{\mathrm{ANN,v}} = 20.82\%$ for particle velocities and $\mathrm{MAPE}_{\mathrm{ANN,T}} = 18.77\%$ for particle temperatures.

Regarding the PINN models, the derivatives appearing in the physics-informed loss term are approximated based on the available data sets. We compare the values on the left and right sides of the above-mentioned equations, aiming to assess the congruence of the data sets with the equations. The computed values for both the left and right sides of the equations used in PINN velocity and temperature models are visualized in Fig. 4(a) and (b), respectively. The x-axis in this figure corresponds to the 45 conducted simulations. The values on the y-axis represent the averaged values of the left and right sides of the equations across all particle trajectories in the respective simulation at an example stand-off distance of d = 150 mm. From Fig. 4(a), it is evident that the available training data sets from the simulations align very well with the corresponding equation formulated in Eq 3 for particle velocity, with MSE of about 0.03%. In contrast, Fig. 4(b) shows that the data sets do not conform well to the equation formulated in Eq 5 for particle temperature, with MSE of about 133.93%. One possible reason could be that the approximation of temperature derivative is not precise enough if the particle temperature drastically varies during the simulation. Another reason is that the influencing factors on particle temperatures are much more complex than those on particle velocities. Factors such as radiation, heat transfer, particle vaporization, and other physicochemical mechanisms play dominant roles in temperature dynamics. The results suggest that the complete system of governing

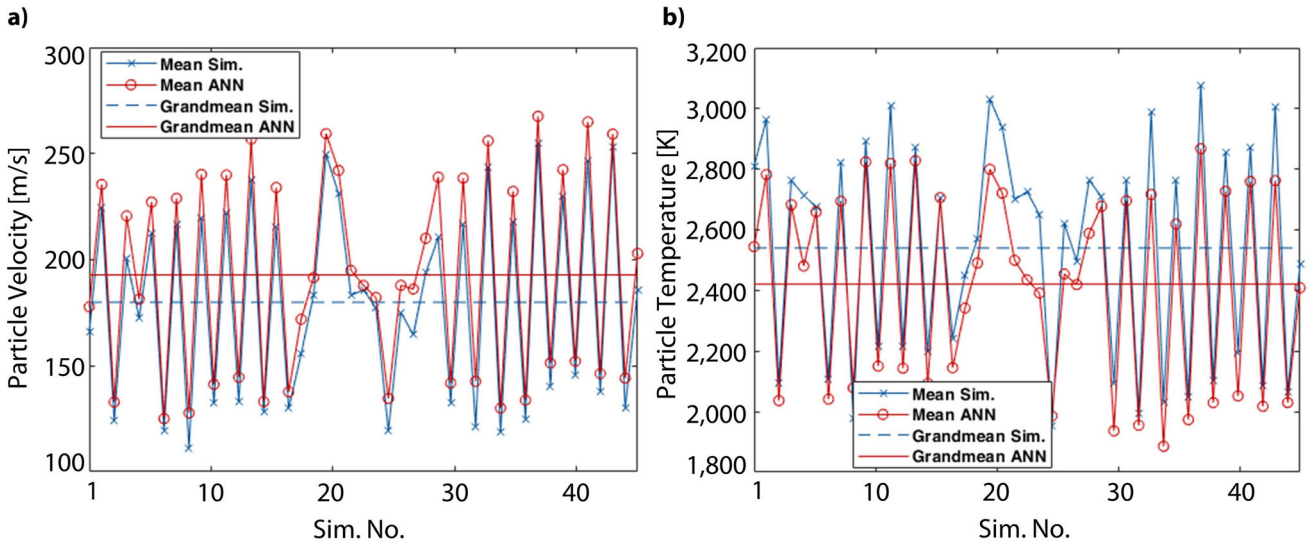

Fig. 3 Results of the mean particle (a) velocities and (b) temperatures per simulation for ANN models

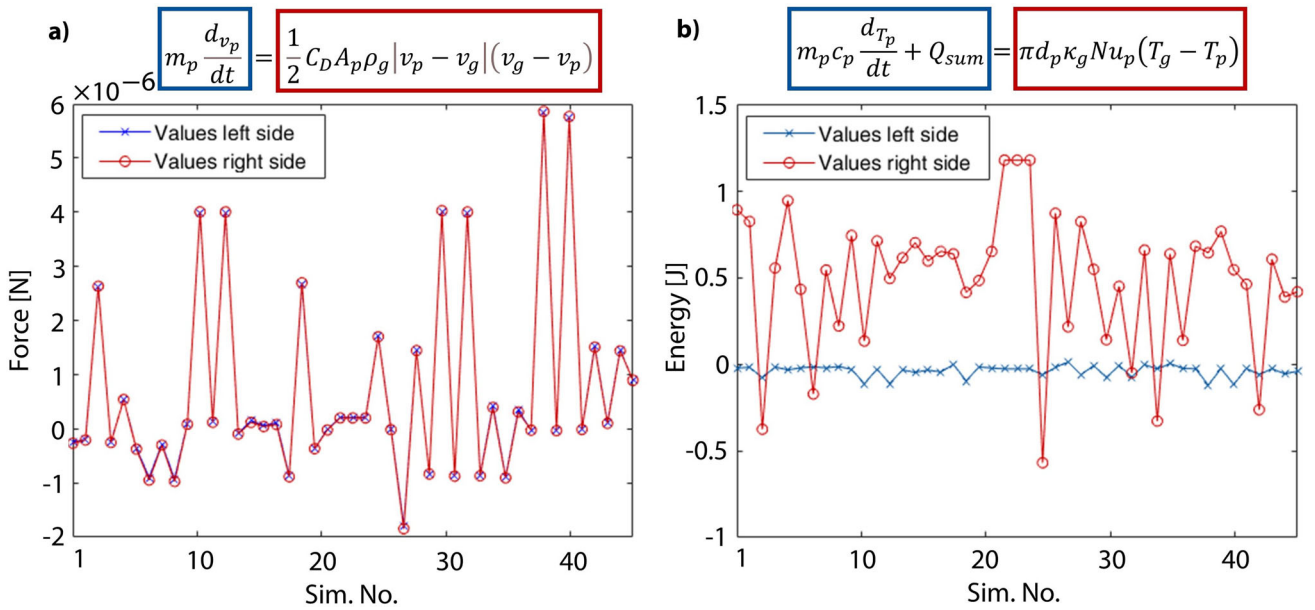

Fig. 4 Comparison of the left and right sides of the equations used in PINN models for (a) velocity and (b) temperature for each of the 45 simulations

equations for particle temperature, and not just a subset of them, is essential for accurate PINN models. Moreover, it is noteworthy that the difference between both sides of the equation for particle velocity in Fig. 4(a) is relatively small. Thus, a large weight parameter (α) for the L_{PDE} would be desirable for velocity modeling.

Analogously, the results of the mean particle a) velocities and b) temperatures per simulation from the PINN models are shown in Fig. 5. The MAPE values for individual particles are calculated to be $\mathrm{MAPE}_{\mathrm{PINN,v}} = 20.06\%$ for particle velocities and $\mathrm{MAPE}_{\mathrm{PINN,T}} = 19.45\%$ for particle temperatures. A precise prediction of the properties of each single particle is barely expected with the ML methods at hand due to the randomness of particle behavior caused by turbulence in the plasma flow and particle collisions. However, accurately predicting average particle properties remains a key performance indicator in plasma spraying and is valuable for investigating the interrelationships between process parameters and coating properties. The results indicate that the PINN models can precisely predict the trend of mean particle properties. Furthermore, they show that particle velocity can be predicted more accurately than particle temperature, which is

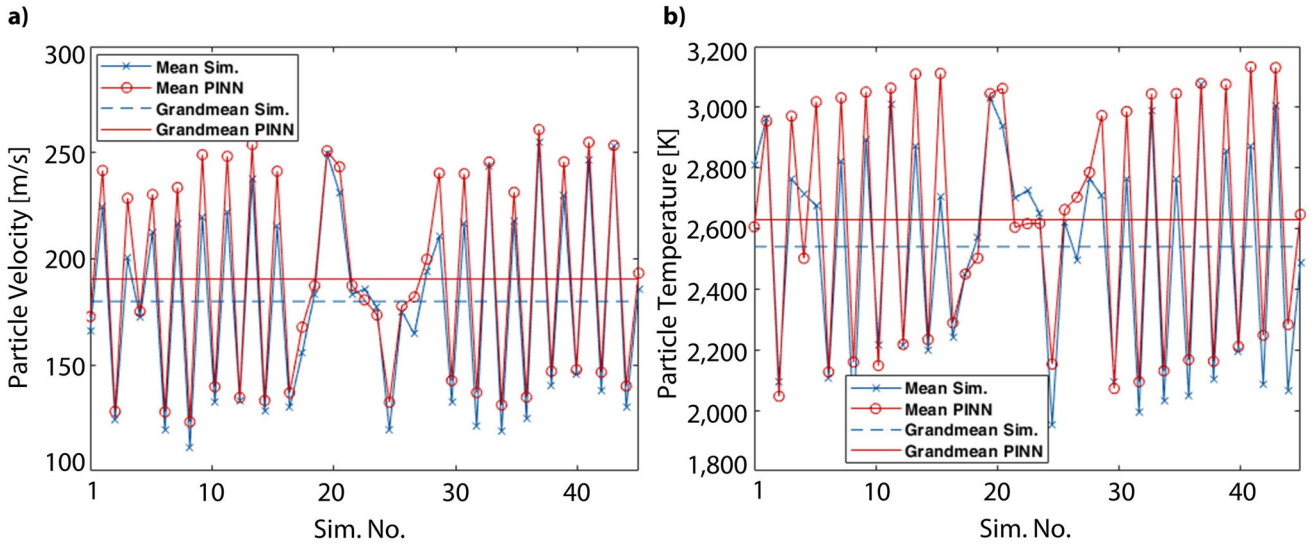

Fig. 5 Results of the mean particle (a) velocities and (b) temperatures per simulation for PINN models

due to the high complexity of the heat transfer equation system in the simulations.

When doing a comparison, the results show that the PINN model performs better than the corresponding ANN model in predicting particle velocity, as evidenced by improvements in the MAPE values. While the enhancement may not be substantial, it suggests that the incorporated equations have the potential to enhance the prediction accuracy of ANN. Additionally, the training of PINN required fewer computational costs (number of epochs) compared to ANN, as the incorporation of physical laws facilitates the learning process. The evaluation of ANN and PINN for temperature prediction reveals a different outcome, with the performance of PINN being inferior to that of ANN. This discrepancy could be attributed to the intricacies involved in heat transfer and particle temperature behavior. Another contributing factor could be not considering the comprehensive equation systems covering the entire fluid domain in our study, potentially leading to a limited representation of the underlying physics. Furthermore, the complex interplay of variables within the heat transfer process might pose challenges that the PINN model struggles to capture effectively.

Summary and Outlook

This work represents an initial application of physics-informed neural networks (PINNs) to the APS process, introducing their potential advantages to the field. The primary goal of this study was to harness the potential of PINNs to enhance the accuracy of predicting particle properties using simulation data sets, thereby circumventing the computational expenses associated with CFD simulations. Upon analyzing the results for predicting particle velocity and temperature using both ANN and PINN models, it was observed that PINN outperforms ANN in predicting particle velocity. However, predicting particle temperature remains challenging, indicating the need for further enhancements in the PINN model.

A potential approach for enhancing the PINN model developed in this study is to integrate additional equations into the physics-informed loss term. Specifically, introducing a system of governing PDEs such as the Navier–Stokes equations with corresponding boundary conditions could significantly improve the accuracy of the model. To facilitate this improvement, it is crucial to export not only data on particle trajectories but also the complete spatiotemporal fluid field within the plasma jet. This expansion allows the model to capture the turbulent properties of the plasma gas, which were not fully exploited in this study. Simultaneously, a balance between the computational cost for training the PINN model with a comprehensive system of PDEs and that of CFD simulations should be carefully considered.

Acknowledgments Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy-EXC-2023 Internet of Production-390621612. Simulations were performed with computing resources granted by RWTH Aachen University under project rwth1204.

References

1. K. Bobzin and M. Öte, Modeling Plasma-Particle Interaction in Multi-Arc Plasma Spraying, *J. Therm. Spray Technol.*, 2017, **26**, p 279–291.
2. M. Li and P.D. Christofides, Computational Study of Particle In-Flight Behavior in the HVOF Thermal Spray Process, *Chem. Eng. Sci.*, 2006, **61**, p 6540–6552.
3. J.P. Trelles, C. Chazelas, A. Vardelle and J.V.R. Heberlein, Arc Plasma Torch Modeling, *J. Therm. Spray Technol.*, 2009, **18**, p 728–752.
4. A.F. Kanta, G. Montavon, M. Vardelle, M.P. Planche, C.C. Berndt and C. Coddet, Artificial Neural Networks Versus Fuzzy Logic: Simple Tools to Predict and Control Complex Processes—Application to Plasma Spray Processes, *J. Therm. Spray Technol.*, 2008, **17**, p 365–376.
5. M.A.R. Mojena, A.S. Roca, R.S. Zamora, M.S. Orozco, H.C. Fals and C.R.C. Lima, Neural Network Analysis for Erosive Wear of Hard Coatings Deposited by Thermal Spray: Influence of Microstructure and Mechanical Properties, *Wear*, 2017, **376–377**, p 557–565.
6. J. Zhu, X. Wang, L. Kou, L. Zheng and H. Zhang, Prediction of Control Parameters Corresponding to in-Flight Particles in Atmospheric Plasma Spray Employing Convolutional Neural Networks, *Surf. Coat. Technol.*, 2020, **394**, 125862.
7. K. Bobzin, W. Wietheger, H. Heinemann, S.R. Dokhanchi, M. Rom and G. Visconti, Prediction of Particle Properties in Plasma Spraying Based on Machine Learning, *J. Therm. Spray Technol.*, 2021, **30**, p 1751–1764.
8. K. Bobzin, H. Heinemann, S.R. Dokhanchi and M. Rom, Replication of Particle Trajectories in the Plasma Jet with Two Consecutive Residual Neural Networks, *J. Therm. Spray Technol.*, 2023, **32**, p 1447–1464.
9. S. Cuomo, V.S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi and F. Piccialli, Scientific Machine Learning Through Physics-Informed Neural Networks: Where we are and What's Next, *J. Sci. Comput.*, 2022, **92**, p 88.
10. S. Cai, Z. Mao, Z. Wang, M. Yin and G.E. Karniadakis, Physics-Informed Neural Networks (PINNs) for Fluid Mechanics: A Review, *Acta Mech. Sin.*, 2021, **37**, p 1727–1738.
11. L. Gui, B. Wang, R. Cai, Z. Yu, M. Liu, Q. Zhu, Y. Xie, S. Liu and A. Killinger, Prediction of In-Flight Particle Properties and Mechanical Performances of HVOF-Sprayed NiCr-Cr_3C_2 Coatings Based on a Hierarchical Neural Network, *Materials*, 2023, **16**, p 6279.
12. Z. Li, F. Liu, W. Yang, S. Peng and J. Zhou, A Survey of Convolutional Neural Networks: Analysis Applications, and Prospects, *IEEE Transact. Neural Netw. Learn. Syst.*, 2022, **33**, p 6999–7019.
13. H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, Recent Advances in Recurrent Neural Networks, 2018
14. M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations, 2017
15. M. Raissi, P. Perdikaris and G.E. Karniadakis, Physics-Informed Neural Networks: A Deep Learning Framework For Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations, *J. Comput. Phys.*, 2019, **378**, p 686–707.
16. M. Öte, Understanding multi-arc plasma spraying *Dissertation* Rheinisch-Westfälische Technische Hochschule Aachen and Shaker Verlag
17. K. Bobzin and M. Öte, Modeling Multi-Arc Spraying Systems, *J. Therm. Spray Technol.*, 2016, **25**, p 920–932.
18. A. Burcat, B. Ruscic, Chemistry, Third Millenium Ideal Gas and Condensed Phase Thermochemical Database for Combustion (with Update from Active Thermochemical Tables), 2005
19. AG. Baydin, B.A. Pearlmutter, A.A. Radul, J.M. Siskind, Automatic Differentiation in Machine Learning: A Survey, 2015