Journal of Manufacturing Processes 165 (2026) 461–471

Contents lists available at ScienceDirect

Journal of Manufacturing Processes

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

Integrating computational fluid dynamics and artificial intelligence for predicting in-flight thermo-kinetic properties in cold spray☆

Roberta Falco [a], Kirsten Bobzin [b], Hendrik Heinemann [b], Marvin Erck [b], Kevin Jasutyn [b], Christopher Wasels [b], Sara Bagherifard [a,*]

[a] Department of Mechanical Engineering, Politecnico di Milano, Italy
[b] Surface Engineering Institute, RWTH Aachen University, Germany

HIGHLIGHTS

- Cold Spray repair is modeled under varied process and substrate conditions
- Fluid dynamics simulations accurately predict particle speed and temperature
- Support Vector Regression builds fast predictive models for particle properties
- Explainable AI grants interpretability of factors most affecting particle behavior
- Combined modeling and AI enable efficient Cold Spray process optimization

ARTICLE INFO

Keywords:
Cold spray
Solid-state additive manufacturing
Computational fluid dynamics
Process modeling
Machine learning
Explainable artificial intelligence

ABSTRACT

Cold Spray is a solid-state deposition technique that accelerates metallic particles to supersonic speed through a converging-diverging nozzle. Particles adhere to the substrate thanks to the high kinetic energy they acquire when reaching a material-specific critical velocity. The optimal choice of process parameters — essential to obtain suitable particle temperatures and velocities — is crucial for deposition efficiency. In the present study, a computational fluid dynamic (CFD) model was developed to simulate the cold spraying process. The model was validated experimentally through a high-speed camera for in-flight particle tracking. The simulations were repeated on a wide range of process parameters and on different substrate geometries, applying Latin Hypercube sampling to ensure a homogeneous distribution within the variable space. The generated data was used to train an artificial intelligence (AI) model of Support Vector Regression (SVR) with the objective of directly predicting the thermo-kinetic properties of the metallic powders. To strengthen the interpretability of the prediction model, explainable AI method of SHapley Additive exPlanations (SHAP) was implemented to identify how each input parameter affects the model predictions for particle temperatures and velocities. The combined CFD-AI approach showed high accuracy and efficiency in predicting the thermo-kinetic conditions of the powder while maintaining the physical interpretability of the related phenomena. This integrated method enables advanced optimization strategies for controlling the Cold Spray process.

1. Introduction

Cold Spray (CS) is a solid-state deposition technique in which micrometric metallic or composite particles are propelled to supersonic velocities—typically between 500 and 1200 m/s—via a high-pressure, inert gas flow passing through a converging-diverging DeLaval nozzle [1]. Unlike other thermal spray processes, CS maintains particle temperatures well below their melting points, preventing undesirable thermal effects such as oxidation, phase transformation, or heat-affected zones. Instead of melting, coating formation relies entirely on the kinetic energy of the particles. Successful particle bonding is obtained when they surpass a critical velocity, dependent on the temperature and thermomechanical properties of the sprayed material [1,2].

Originally developed for coatings, CS is now gaining traction also in

☆ This article is part of a Special issue entitled: 'AI for Intelligent Welding' published in Journal of Manufacturing Processes.

* Corresponding author.
E-mail address: sara.bagherifard@polimi.it (S. Bagherifard).

https://doi.org/10.1016/j.jmapro.2026.03.030
Received 8 October 2025; Received in revised form 1 February 2026; Accepted 13 March 2026

additive manufacturing (AM) and repair applications [3,4]. Its solid-state nature makes it especially advantageous for eliminating heat-affected zones (HAZ) and repairing temperature-sensitive substrates. This applies in particular when compared to commonly used melting-based repair techniques such as welding or brazing. Other solid-state processes, such as linear friction welding and friction stir welding [5,6], while avoiding melting and preserving consistency, are limited in geometry and require tool access Further benefits of CS include high deposition rates, minimal oxidation, and compatibility with geometrically complex or oxygen-sensitive components [7].

The optimization of CS parameters, such as gas pressure, temperature, nozzle design, and standoff distance (SoD), requires a deep understanding of the interaction between supersonic gas flows and particle dynamics. In this regard, Computational Fluid Dynamics (CFD) is a powerful tool, enabling detailed simulation of gas flows, particle trajectories, and thermal histories. Specifically, CFD models have been extensively used to study how different nozzle geometries and operational conditions influence the supersonic flow of the carrier gas [8–10]. For instance, higher gas stagnation temperatures, increased stagnation pressures, or the use of lighter carrier gases like helium are known to raise particle velocities, either by boosting gas speed or density in and beyond the nozzle [11–13].

However, two key limitations remain. First, most CFD investigations assume planar substrates, whereas real repair scenarios often involve cavities, curved surfaces, or complex geometries that significantly alter particle trajectories, impact angles, and local bow-shock structures [12,14]. Second, while high-fidelity CFD models can capture these effects, they are computationally expensive and impractical for exploring the high-dimensional parameter space associated with CS, especially when multiple process variables and geometric descriptors are considered simultaneously.

To address this challenge, the integration of Artificial Intelligence (AI) methodologies offers a promising avenue. By training machine learning (ML) models on data generated from CFD simulations, it becomes possible to construct surrogate models that can accurately and efficiently predict the thermo-kinetic properties of particles without the need for repeated full-scale simulations. These models can serve as rapid evaluators, enabling near-instantaneous estimation of particle behavior across a wide range of conditions, thereby significantly accelerating the design and optimization loop for the CS process [15–17]. In CS, however, such approaches remain scarce. To date, the only reported CFD-based surrogate modeling framework for CS is the work of Mehmood et al. [15], which focused primarily on nozzle geometry optimization and did not address particle-substrate interaction or substrate geometry effects. More broadly, existing ML applications in CS have largely targeted empirical predictions of deposition efficiency or porosity [18,19], often without direct coupling to CFD simulations or without resolving particle thermo-kinetic states.

Moreover, despite their predictive power, AI models are often criticized for their “black-box” nature, which limits their adoption in domains where physical interpretability and trust are paramount [20]. In the context of CS, it is not sufficient to merely predict particle velocities and temperatures; it is equally important to understand *why* certain input parameters exert more influence than others and how they interact with the underlying physical principles of the process.

To address this gap, we introduce a hybrid framework that combines CFD, ML, and Explainable AI (XAI) techniques to enable both accurate prediction and interpretable analysis of particle dynamics in CS. This approach builds on a previous XAI study [21] and ML model based on Support Vector Machine [22] in atmospheric plasma spraying. Specifically for XAI, we employ SHapley Additive exPlanations (SHAP) [23] to assess the relative influence of input features on ML model outputs. By quantifying feature importance, SHAP enhances the physical interpretability of data-driven models, enabling not only the prediction outcomes but also a deeper understanding of the causal relationships between processing conditions and deposition behavior.

In this work, a high-fidelity CFD model is used to simulate particle velocity and temperature for varying gas temperature, gas pressure, particle diameter, cavity depth, cavity radius, powder feed rate, and stand-off distance. A carefully designed design of experiments allows this high-dimensional input space to be efficiently sampled using only 40 CFD simulations. The resulting dataset is then used to train ML surrogate models capable of predicting particle velocities and temperatures across the design space with negligible computational cost. Crucially, SHAP analysis is subsequently employed to interpret the surrogate model predictions, providing quantitative insight into the relative importance and interaction of process parameters and geometric features. This enables not only rapid prediction but also physically meaningful interpretation of how gas dynamics, particle characteristics, and substrate geometry jointly influence particle acceleration and heating. This integrative approach offers a robust and scalable pathway to accelerate CS process optimization, reduce reliance on computationally expensive simulations, and support real-time decision-making in advanced manufacturing and repair.

2. Methodology

2.1. CFD modeling of the cold spray process

The analysis of flows through variable-area ducts is a common research problem in various fields. As such, analytical formulations have traditionally been applied to describe such gas flows under set assumptions. One of the simplest and widely used gas flow models is developed under the assumption of one-dimensional (1D) isentropic ideal gas flow [24]. The 1D model can be a good first approximation, as it is sufficiently accurate and simple to apply. On the other hand, it does not consider viscosity, turbulence, boundary layers, nozzle length, or any conditions outside the nozzle, such as bow shocks.

Similarly, particle velocity u_p within a nozzle can be analytically modeled under the assumption of diluted particle presence within the gas and mainly axial velocity, as follows [25]:

$$m_p \frac{\partial u_p}{\partial t} = \frac{1}{2} C_D \left(Re_p\right) \rho_g A_p \left(u_g - u_p\right) \left|u_g - u_p\right| \tag{1}$$

The particle acceleration $\partial u_p/\partial t$, therefore, depends on the difference between gas velocity u_g and particle velocity u_p, particle cross-sectional area A_p and mass m_p, gas density ρ_g and drag coefficient C_D. The latter is, in turn, a function of Reynolds number Re_p dependent on the chosen drag model.

In an analogous way, particle temperature T_p can be described by the following Eq. [25]:

$$m_p c_{p,p} \frac{\partial T_p}{\partial t} = \frac{\nu \lambda_g}{d_p} S_p \left(T_g - T_p\right) \tag{2}$$

Here, $c_{p,p}$ denotes the heat capacity of the particle, ν the heat transfer coefficient, λ_g the thermal conductivity of the gas, d_p the diameter of the particle, and S_p the surface area of the particle.

Eqs. 1 and 2 can be integrated analytically in a straightforward manner only under the assumption of constant gas velocity, temperature, density, and drag coefficient across the divergent section of the nozzle. The formulas also do not solve the particle velocity and temperature evolution outside the nozzle, where particles interact with pressure waves.

CFD methodologies, on the other hand, ensure higher fidelity both inside and outside of the nozzle, through the solution of the full Navier-Stokes equations. In this study, the simulations were carried out on OpenFOAM 11. For the sake of computational efficiency, only a 3-dimensional slice of the full axi-symmetrical nozzle was modeled. The nozzle geometry used as a reference was the OUT1 nozzle of Impact Innovations (DE), with nitrogen as a carrier gas.

The pre-chamber, convergent, throat, and divergent sections of the

nozzle, as well as the external domain, were defined as distinct blocks through the *blockMesh* function. The external dominium, in particular, was delimited by the substrate cavity, which presented a spherical cap indentation of set radius and depth, to model typical repair defect preparation geometries. The mesh generation was automated through a Python code, to allow for the parametrization of SoD and defect geometry. The parametrization also allows for proper refinement of the mesh close to the nozzle walls and on the interface with the substrate, while maintaining continuity and near-square cells throughout the model. The *mergeMesh* and *stitchMesh* functions were also applied to this same purpose.

The simulations were carried out in three consecutive steps to ensure an accurate representation of the process, as well as a limited computation cost. The aim of the first step is for the gas flow to reach steady-state conditions. The *fluid* solver module was applied, which is suitable for transient turbulent flow of compressible fluids, with a high Courant number and more diffusive and stable numerical schemes. This setup allows for a fast first-approximation solution. The second step maps the results of the first as initial condition and implements the *shockFluid* solver for a density-based solution of the compressible flow. In this case, higher-order, more accurate numerical schemes were set, lowering the Courant number to avoid divergence. The combination of these two steps allows for an accurate description of the gas flow, limiting computational effort and divergence.

The third step is where particle tracking is introduced. The *thermoCloud* Lagrangian particle model was selected to model the heat exchange from the fluid gas stream to the solid particles. The drag model described by Plessis and Masliyah [26] was selected, as it was deemed to be the best fit to CS deposition in previous studies [27]. In OpenFOAM, the only available heat transfer model for gas-particle interactions is the *RanzMarshall* model, based on the work of Ranz and Marshall [28]. This model defines the heat transfer coefficient ν using the following correlation:

$$\nu(Re_p) = 2 + 0.6Re_p^{0.5}\mathrm{Pr}^{0.33} \tag{3}$$

Here, Pr is the Prandtl number of the gas, given by $Pr = \mu_g c_{p,g}/\lambda_g$, where μ_g is the gas dynamic viscosity, $c_{p,g}$ is the gas specific heat capacity, and λ_g is the thermal conductivity of the gas.

The particles were injected from the inlet patch with conical injection, according to the selected powder feed rate. The particle size distribution was set based on the cumulative size probability distribution of pure Cu ($\geq$99.9%) powder supplied by Safina A.S (Czech Republic). Default thermophysical properties are not available for copper; therefore, a custom source file was created. The specific heat of copper was modeled with the standard NASA 7-coefficient polynomial form. Coefficients were obtained from the NIST–JANAF Shomate data for solid copper (298–1358 K) [29]. Particles were set to be able to rebound off the nozzle walls, while interactions among particles were deemed negligible due to the low particle concentration within the gas flow [30].

For each simulation, the particle injection time was such to guarantee the injection of around 500 particles, for better data distribution. The particle velocities and temperatures at the impact with the substrate were therefore extracted.

To investigate the influence of each of the input parameters, multiple simulations need to be run within their typical working ranges. Due to the computational effort required for the CFD simulations, the total number of runs was limited to 40 trials. Therefore, to efficiently explore the large multidimensional design space, a Latin Hypercube Sampling (LHS) strategy [31,32] was adopted. The selected ranges for each parameter are listed in Table 1, while Fig. 1 shows the pair plots highlighting the uniformity of trial distributions within such ranges.

2.2. Experimental validation

The validity of the CFD approach was assessed through a three-fold comparison: with analytical models based on fluid dynamics calculation, with the Kinetic Spray Solutions (KSS) software (Buchholz, Germany) [33], and with experimental particle velocity acquisitions during spray sessions. The calculations in the KSS software are based on a combination of the isentropic theoretical model with data fitted from CFD simulations to adapt to different processing conditions [34,35].

Concerning the gas evolution inside and outside the nozzle, the values of gas temperature, velocity, and pressure along the nozzle axis were extracted from the simulation and compared to both the isentropic analytical formulation [24] and the results extracted from the simulation tab of the KSS software. The particle velocity was instead validated experimentally with the HiWatch HR particle velocimetry system (Oseir Ltd., Finland). Impact Innovations 6/11 CS system with OUT 1 nozzle was employed to spray pure Cu powder with the particle size distribution presented in Table 2 and the process parameters listed in Table 3.

Table 1
Data input from CFD simulations for ML model training.

Parameter	Interval	Unit
Gas temperature	600–800	°C
Gas pressure	35–50	bar
Particle Diameter	14–52	µm
Cavity depth	0.1–5.0	mm
Cavity radius	5.1–9.9	mm
Powder feed rate	48–125	g/min
Stand-off distance	20–40	mm

2.3. Machine learning models

The data from the CFD simulations is used to train ML models, including Support Vector Regression (SVR) and Artificial Neural Networks (ANN). Each model is trained to predict one of the two particle properties at the position of the substrate: the particle velocity component in spray direction v_x and the particle in-flight temperature T_P. The inputs of these models are listed in Table 1.

Despite having identical input parameters, the particle in-flight properties vary significantly due to differences in their trajectories within the gas jet. The models aim to predict the average behavior of particles rather than exact values. A small fraction, 2.5% of the simulated particles, showed unphysical behavior, such as velocities below 100 m/s, and were therefore excluded from the dataset. The objective is to approximate a function $f(x)$ that maps input vectors x to target values, either T_P or v_x. A total of 40 CFD simulations generated approximately 19,000 data points. In each simulation, all input parameters remain constant for every particle, except the particle diameter, which follows the real particle distribution. Thus, each particle yields one sample of the target function. For the model fitting, the dataset was split: 80% of the simulations are used for training, and the remaining 20% is used for testing. Both ANN and SVR show similarly high prediction accuracy. However, the ANN requires a large number of neurons (350 neurons with one hidden layer for particle velocity prediction and 550 neurons with two hidden layers for particle in-flight temperature prediction), resulting in significantly higher computational costs compared to SVR. On the same CPU-based training setup, the SVR model required approximately 8 s for training, whereas the ANN required about 200× longer training time. Therefore, the following section focuses solely on the SVR approach applied in this study.

2.3.1. Support vector regression

SVR is an extension of Support Vector Machines (SVM), introduced by Cortes and Vapnik [36]. It is a supervised learning method for continuous prediction. Unlike standard regression models, SVR fits a linear model to the data while allowing deviations up to a fixed threshold ϵ. The SVR model is defined as:

$$f(x) = w, \phi(x) + b \tag{4}$$

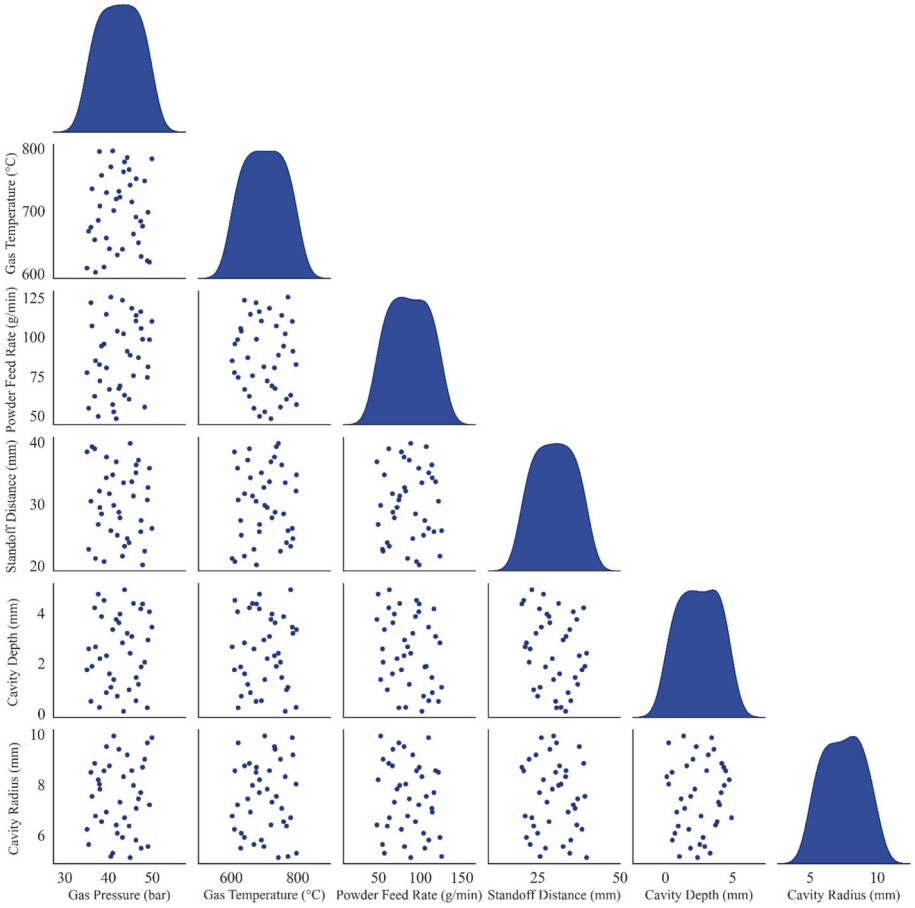

Fig. 1. Pair plot of the distribution of each input parameter among different simulation trials generated with LHS.

Table 2
Cu powder particle size distribution.

Powder	D_{10} (μm)	D_{50} (μm)	D_{90} (μm)
Pure Cu	19	28	42

Table 3
CS process parameters for experimental validation.

Gas Pressure (MPa)	Gas Temperature (°C)	Powder Feed Rate (g/min)	Standoff distance (mm)
5	750	20.75	30

Here, w is the weight vector, b is the bias term, and $\phi(x)$ is a feature transformation. The model is trained on N input-target pairs (x_i, t_i), where each x_i is a vector of input parameters and t_i is the corresponding target value. To limit prediction error, SVR uses a margin of size ϵ:

$$|f(x) - t| \leq \epsilon \tag{5}$$

This defines the ϵ-tube around the predicted function, where deviations up to ϵ are allowed without penalty. The margin boundaries define a region in which the predictions are considered accurate. However, exact adherence to this constraint is usually not possible, especially when the training data contains outliers or noise.

To address this, slack variables ξ and ξ^* are introduced. These variables measure the degree to which predictions fall outside the ϵ-tube. This leads to a soft margin formulation, which permits some errors while keeping the model complexity low. The optimisation problem of the SVR in its primal form becomes [37]:

$$Minimize: \frac{1}{2}w^2 + C\sum_i (\xi_i + \xi_i^*) \tag{6}$$

Subject to:

$$f(x) - t \leq \epsilon + \xi_i^* \tag{7}$$

$$t - f(x) \leq \epsilon + \xi_i \tag{8}$$

$$\xi_i, \xi_i^* \geq 0 \quad \forall i = 1, \ldots, N \tag{9}$$

Here, C is a regularisation parameter that controls the trade-off between model complexity and tolerance for errors. Minimizing w^2 ensures flatness of the regression function, discouraging overly complex models. The formulation is convex, ensuring a unique global solution. From the primal problem, a dual formulation is derived using Lagrange multipliers α. The solution has the form [38]:

$$f(x) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*)\phi(x_i), \phi(x) + b \tag{10}$$

Only the samples that lie on or outside the ϵ-tube contribute to this sum. These are the support vectors. The Karush-Kuhn-Tucker conditions ensure that all other training samples have zero Lagrange multipliers and do not influence the model directly [38].

In many cases, the relationship between input and output is not linear in the original feature space. To address this, the kernel trick is used. A kernel function replaces the dot product in (Eq. 10):

$$k(x_i, x) = \phi(x_i), \phi(x) \tag{11}$$

This allows the model to operate as if it were in a higher-dimensional

feature space, without explicitly computing $\phi(x)$. The resulting prediction function becomes:

$$f(x) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) k(x_i, x) + b \tag{12}$$

In this study, the radial basis function (RBF) kernel is used. It is defined as:

$$k(x_i, x_j) = \exp\left(- \frac{x_i - x_j^2}{2\gamma^2} \right) \tag{13}$$

The RBF kernel measures similarity between samples. The hyperparameter γ controls the spread of the kernel. By applying the kernel function, the SVR operates as if it were fitting a linear model in an infinite-dimensional feature space. However, the feature transformation $\phi(x)$ does not need to be computed explicitly [38].

Before optimization, the input data is standardised. This is necessary for the RBF kernel, which assumes inputs are centred around zero and have comparable variance. Each input feature is scaled to zero mean and unit variance:

$$x_{,\ i} = \frac{x_{unscaled.i} - \mu_i}{\sigma_i} \tag{14}$$

Here, μ_i and σ_i are the mean and standard deviation of feature i. In this study, the implementation of the SVR from the scikit library was used. The following hyperparameters listed in Table 4 are optimized using Bayesian optimization.

2.4. Explaining machine learning models with Shapley values

Shapley values originate from cooperative game theory [39]. In this context, a group of players collaborate to earn a reward. The Shapley value quantifies each player's contribution by measuring how the reward changes when that player joins the group.

In ML, the prediction process is often difficult to interpret. Even in white-box models, it is challenging to determine the individual contribution of each input feature x_i to the model output $f(x)$. To address this, an explanation model g is used to approximate $f(x)$ as a linear combination of feature contributions:

$$g(f, x) = \sum_{i=0}^{M} \phi_i(f, x) \tag{15}$$

Here, M is the number of input features, and $\phi_i \in \mathbb{R}$ represents the contribution of each feature. This additive model is easier to interpret and aligns with human reasoning.

SHapley Additive exPlanations (SHAP) adapt Shapley values to explain model outputs. This method was introduced in [40] and briefly presented in this section. For a given input x, the SHAP value ϕ_i quantifies the contribution of each feature to the prediction $f(x)$. Analogous to players in a game, each feature may or may not participate in the prediction. The simplified input $z^{'}$ belongs to the set Z of all possible feature subsets.

Some models, such as SVR, do not support missing features. In such cases, missing features are handled by computing conditional expectations based on background data:

$$f(z') = E[f(z) \,|z_S] \tag{16}$$

Here, z_S denotes the subset of features included in the prediction.

Table 4
Hyperparameter search spaces.

Model Parameter	Search Space
Regularisation C	1e+1 to 1e+6
Tolerance ϵ	1e-6 to 1e-2
Kernel variance γ	1e-4 to 1e+1

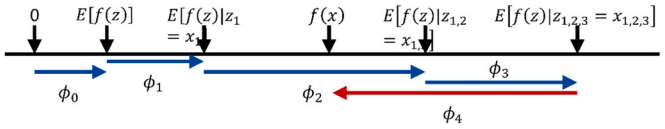

Fig. 2. Explanation of a model output using SHAP values, adapted from [40].

Fig. 2 illustrates this process, showing how the output is reconstructed as a combination of such conditional expectations. Because input features are often dependent, SHAP values are computed by averaging over all feature orderings, leading to exponential complexity $\mathcal{O}(2^M)$.

The base value ϕ_0 represents the expected model output when no features are known. The SHAP values are the solution of the following equation:

$$\phi_i(f, x) = \sum_{z' \in Z} \frac{|z'|!(M - |z'| - 1)!}{M!} \left(\mathrm{E}[f(z) \,|z_S] - \mathrm{E}[f(z) \,|z_S, z_i] \right) \tag{17}$$

The kernel method introduced in [40] was used to compute the SHAP values. This method uses weighted linear regression to compute the feature contributions. The SHAP values satisfy three important properties:

- Local accuracy: The explanation model matches the original model for the given input.
- Missingness: Features with no effect on the prediction receive a SHAP value of zero.
- Consistency: If the model changes, a SHAP value does not change unless the contribution of a feature changes.

3. Results and discussion

3.1. Numerical model validation

Fig. 3 showcases the results of one of the CFD simulations used for training, with an inlet pressure of 37.61 bar and an inlet temperature of 958.3 K. Gas velocity (a), temperature (b), and pressure (c) are displayed throughout the cross-section of the nozzle and of the external domain. Sharp bow shocks can be observed at the nozzle exit, as well as the interaction with the notched substrate.

The values of gas velocity, temperature, and pressure were extracted along the nozzle axis from the nozzle inlet until the prescribed SoD for a corresponding simulation on an unnotched substrate with process parameters listed in Table 3, to match experimental acquisitions. They were then compared with the values obtained with the theoretical isentropic 1D model described in Section 2.1, as well as with the predictions of the KSS software, as shown in Fig. 4. The isentropic model neglects the effects of wall friction and boundary layers, as well as the effect of shock waves at the exit of the nozzle. While the isentropic assumption is reasonably accurate up to the nozzle throat, errors increase significantly in the diverging section. Due to the entropy rise, the pressure of the real gas drops more than the isentropic pressure, and the dissipated energy makes the real temperature lower. Real gas velocity, instead, increases with respect to isentropic velocity, due to a higher total enthalpy converted into kinetic energy [24]. These expected behaviors are reflected in the final CFD results when compared to the isentropic ones.

The KSS simulations do not reflect the same physical behavior consistently. They instead model the interaction between gas and a flat substrate in an approximate manner.

The simulated particle velocities were extracted at the moment of impact with the substrate and compared with the KSS software results, with the theoretical model of Eq. 1, and with the experimental particle diagnostics using HiWatch for the process parameter listed in Table 3, as shown in Fig. 5a. The results showed a good match between CFD-simulated and theoretically calculated velocities, with a mean

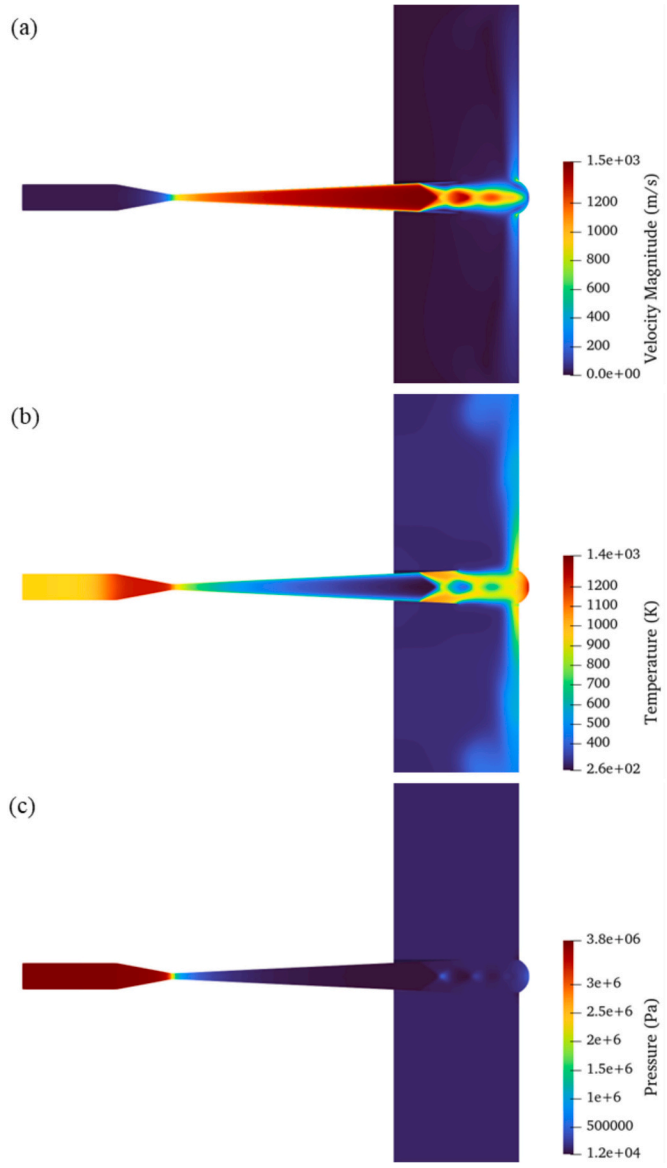

Fig. 3. CFD results for the CS gas evolution on a notched substrate for: (a) velocity, (b) temperature, and (c) pressure.

percentage error (MPE) with KSS software and the isentropic model of 3.6% and 2.0%, respectively, as well as a good match with experimental data.

Due to the well-known difficulty in experimentally measuring single-particle temperatures during CS, a direct experimental validation of particle temperature was not feasible in this work. Consequently, the simulated particle temperature was only compared against KSS results, as shown in Fig. 5b. While this constitutes an indirect validation, KSS is a widely adopted reference tool in the CS community and is itself grounded in validated nozzle flow models. The trend of particle temperatures presents a good match, with an MPE of 6.1%, and the remaining discrepancies are consistent with the previously discussed differences in gas pressure and temperature evolution between KSS and the present CFD model.

This reliance on a CFD-to-CFD comparison represents a limitation of the current study. Nevertheless, the agreement in particle temperature is supported indirectly by the experimental validation of particle velocity and by the accurate reproduction of the gas thermodynamic fields, which largely govern particle heating.

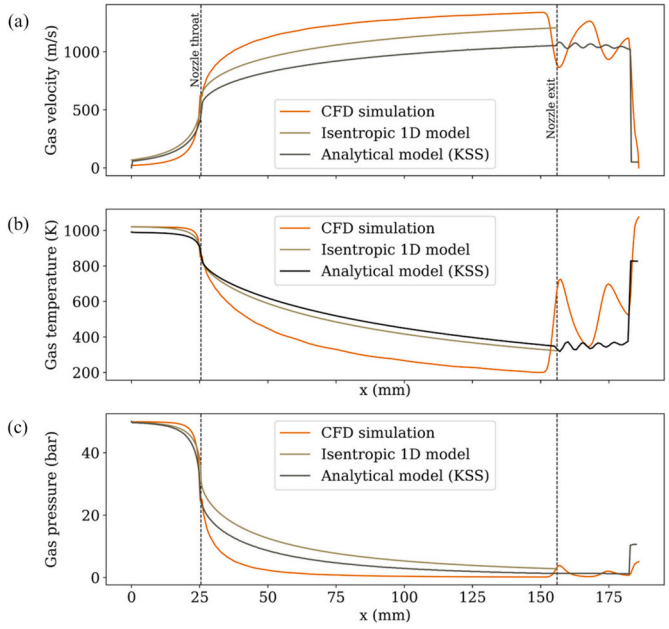

Fig. 4. Plots of the CFD simulation results of gas velocity (a), temperature (b), and pressure (c) along the axis of the nozzle, compared to the theoretical isentropic model and the output of the KSS software.

3.2. Machine learning modeling accuracy

This section presents the results of hyperparameter-tuned SVR models for predicting particle velocity and in-flight temperature. Prediction accuracy is assessed using two statistical metrics: the mean absolute percentage error (MAPE) and the coefficient of determination (R^2). The MAPE measures the average relative error between predictions p_i and simulation targets t_i across N data points, defined as:

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{t_i - p_i}{t_i}\right| \tag{18}$$

Furthermore, the coefficient of determination R^2 quantifies how well the predictions capture the variance in the target values. For N data points, with targets t_i, mean $\bar{t}$, and the prediction p_i, the R^2 score is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(t_i - p_i)^2}{\sum_{i=1}^{N}(t_i - \bar{t})^2} \tag{19}$$

Fig. 6 shows the particle velocity distribution. The black line represents the simulation data, while the yellow dashed line shows the SVR prediction. Both training and test sets form a smooth, unimodal distribution. The prediction closely matches the CFD simulations with a high R^2 value of 0.982 and a low MAPE of 0.89% with root mean square error (RMSE) of just 5.634 m/s.

Fig. 7 shows the temperature distribution. Compared to velocity, the distribution is more complex, exhibiting a noticeable shift between the training and test sets. As a result, prediction accuracy is slightly reduced. The MAPE is 1.15%, and R^2 is 0.882 with RMSE of just 12.659 K, but still indicating a good performance. SVR performs well for both properties, with better results for velocity due to the simpler and more consistent distribution.

A convergence analysis was run to ensure that the dataset size was sufficient to describe the target problem. Figs. 8 and 9, respectively, show the results for velocity and temperature SVM prediction models. Compared to velocity, temperature prediction exhibits higher variance across data splits, reflecting its aforementioned greater physical complexity and sensitivity to local conditions. Despite this, mean test

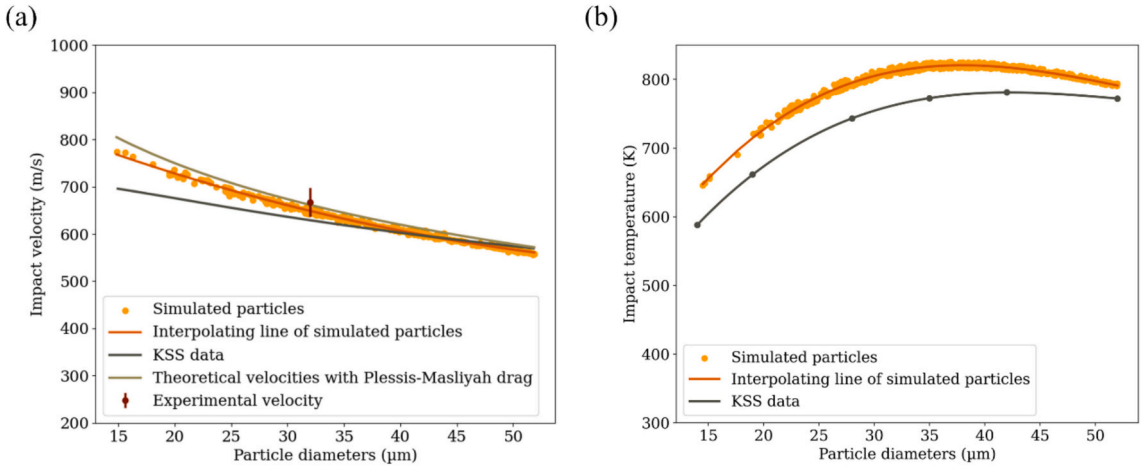

Fig. 5. CFD simulation results of (a) particle impact velocities compared with the KSS software, theoretical and experimental velocities, and (b) particle impact temperatures, compared to KSS software temperatures, for different particle diameters.

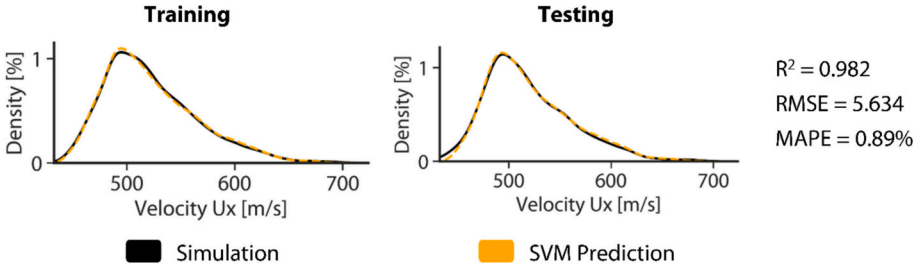

Fig. 6. SVR predictions for the distribution of particle velocities.

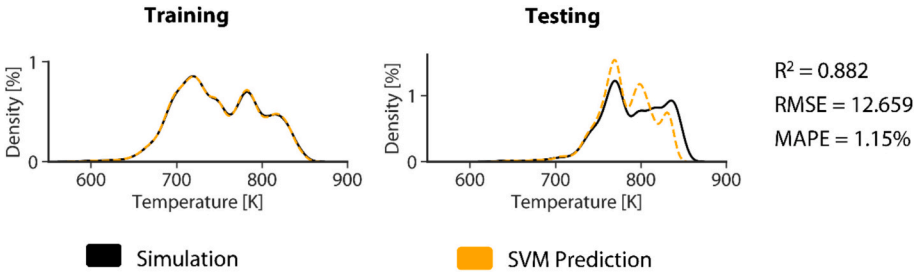

Fig. 7. SVR predictions for the distribution of particle in-flight temperatures.

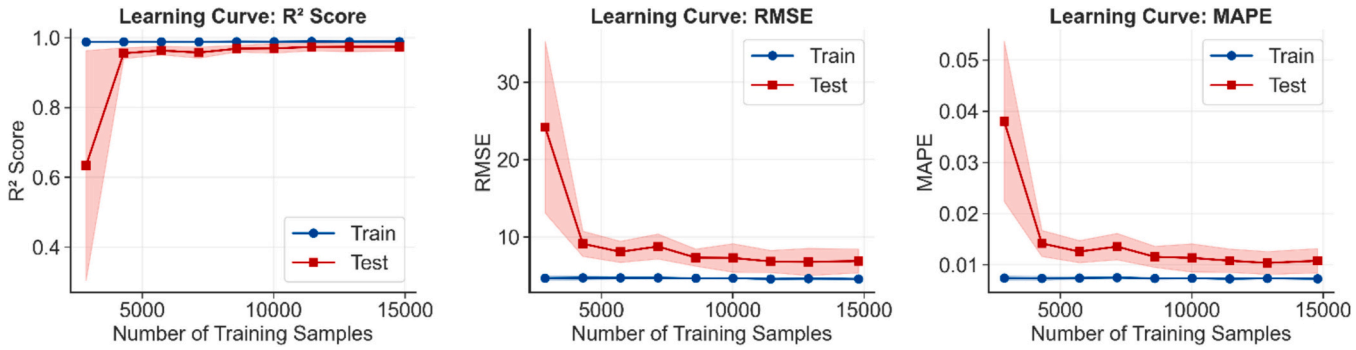

Fig. 8. Convergence analysis for the particle in-flight velocity SVM model.

performance improves consistently with increasing data size, while variance decreases, indicating convergence toward a stable generalization regime.

The optimized hyperparameters used for the SVR are listed in Table 5. For velocity, the model uses a high regularisation parameter C, while the temperature model uses a smaller C and lower ε. This suggests

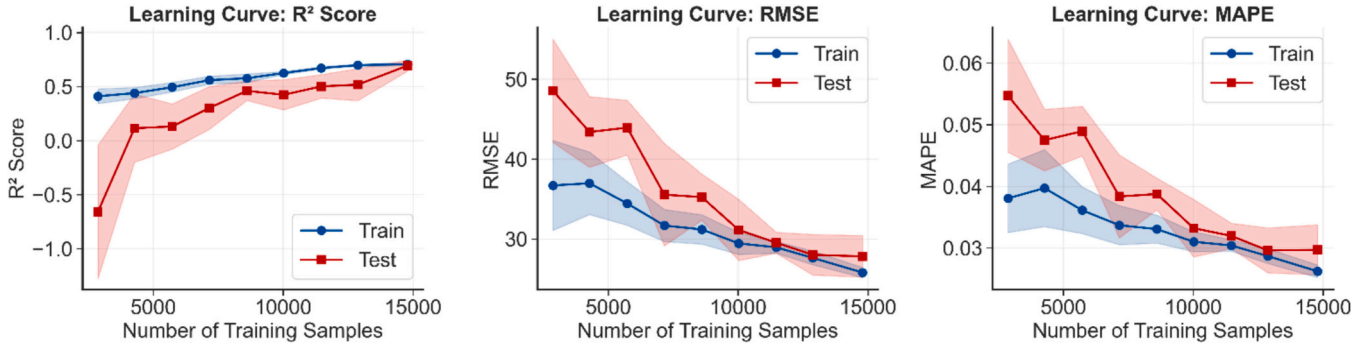

Fig. 9. Convergence analysis for the particle in-flight temperature SVM model.

Table 5
Optimized hyperparameters for the SVR.

Target	Regularisation C	Kernel variance γ	Tolerance ε
Velocity	241,752	1e-2	1e-4
Temperature	1623	7e-3	1e-6

that the velocity data contain less noise, allowing a closer fit. The result is consistent with the distribution complexities observed in Figs. 6 to 9.

The objective of this study, namely the direct prediction of thermo-kinetic properties, has been achieved using ML. The developed models are both efficient and highly accurate and can serve as surrogate models to predict particle velocities and temperatures. While the ultimate objective of this research is to enable reliable repair of different cavity geometries using cold gas–sprayed copper, the present result represents an initial step toward this goal.

3.3. SHAP analysis results

The following section presents the SHAP values calculated for the optimized SVR models that predict particle velocity and in-flight temperature across a test dataset consisting of a total of 3800 data points. The base value for velocity is $v_{x,base} = 491$ m/s, and the base value for the in-flight temperature $T_{p,base} = 730\ K$. These values, computed from the background data, represent the mean predictions.

Fig. 8 shows the SHAP values for particle velocity across the test dataset. Each dot represents a SHAP value for one instance (data point). Each row corresponds to one of the seven input features. The horizontal spread (x-axis) indicates the impact of that feature on the model output. The color represents the feature value: red indicates a high feature value, and blue indicates a low feature value. The vertical density of points reflects the distribution or frequency of SHAP values for that feature. Particle diameter, as well as gas temperature and pressure, are the main contributors to velocity prediction. The results indicate that as the particle diameter decreases, its contribution to the increase in velocity becomes apparent, a trend that is also reflected in the SHAP values. A high particle diameter reduces the predicted velocity by up to −50 m/s from the base value. Powder feed rate, cavity radius, and depth have negligible influence.

Fig. 9 displays the SHAP values for particle temperature across the test dataset. Among all investigated features, gas temperature shows the strongest impact, with higher values leading to increased predicted outputs. Particle diameter also has a significant effect, primarily shifting predictions negatively at lower values. Cavity radius and powder feed rate appear to play only a minor role, with SHAP values centered closely around zero.

The average effect of each input feature on the model output across N data points is given by the mean SHAP values:

$$\overline{\phi} = \frac{1}{N}\sum_{i=1}^{N} |\phi_i| \tag{20}$$

Here, $\overline{\phi}$ is the vector of the mean SHAP values for all input features. Fig. 10 shows the mean SHAP values for the test dataset. Particle diameter and gas temperature are the most influential features for predicting in-flight temperature. For velocity prediction, particle diameter, gas temperature, and gas pressure are the most important, while powder feed rate and cavity radius have little effect. (See Figs. 11 and 12.)

3.4. Discussion

The validation against analytical formulations, commercial software, and experimental measurements confirms the reliability of the CFD model in capturing the main gas-particle interactions in the nozzle. The particle velocity predictions were robust. Also, good correspondence was achieved for the particle temperatures, although the lack of direct experimental measurements limited the validation to comparisons with commercial software. This lack of direct particle temperature measurements remains a broader limitation in CS modeling, highlighting the need for future experimental techniques capable of resolving particle thermal histories in-flight or upon impact.

The surrogate SVR models trained on CFD data demonstrated that high prediction accuracy can be achieved at a fraction of the computational cost of full simulations. The discrepancy in accuracy between the velocity and temperature prediction models reflects the more complex and less uniform distribution of particle temperatures observed in the simulation results, which were found to be strongly affected by gas-particle heat transfer dynamics, dependent on particle size, residence time, and local flow conditions. Nonetheless, both models demonstrated strong potential for practical deployment in CS process optimization.

By employing SHAP, the relative importance of different process

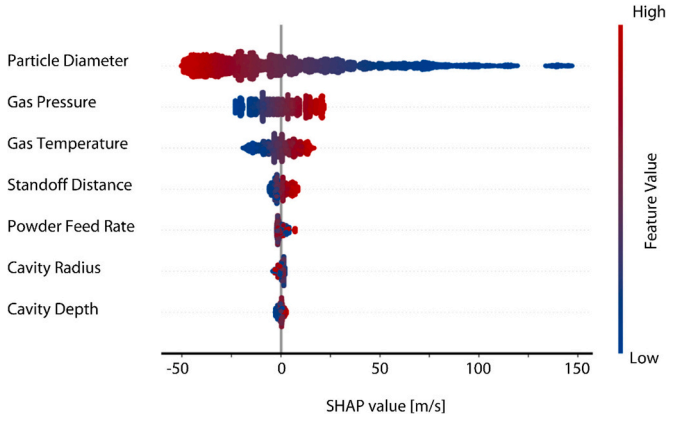

Fig. 10. SHAP values for the particle velocity test data.

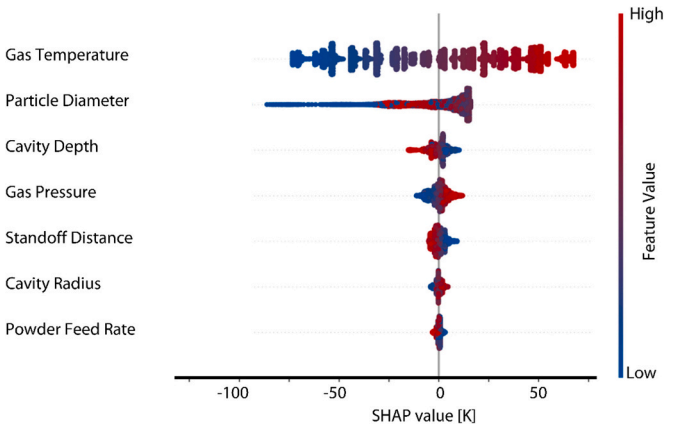

Fig. 11. SHAP values for the particle temperature test data.

parameters was quantified, thus enhancing the interpretability of the ML predictions. The results revealed that particle diameter, gas temperature, and gas pressure exert the strongest influence on particle velocities, while gas temperature and particle diameter dominate the prediction of particle in-flight temperature. These findings are consistent with the established physical intuition and with the CFD simulations themselves, thereby reinforcing confidence in the AI model. Conversely, parameters such as powder feed rate and cavity geometry were found to have negligible effects within the investigated parameter ranges. This suggests that, for certain optimization tasks, the number of relevant process variables could be reduced, simplifying both experimental and numerical design efforts. However, it must be emphasized that SHAP-derived feature importance, represented in the beeswarm plots of Fig. 8 and Fig. 9, is inherently dependent on the sampled input space; extending or shifting the investigated parameter ranges may yield different feature hierarchies.

While the results highlight the robustness of the proposed framework, there are opportunities for further refinement. The defect geometries explored in this study were intentionally simplified (spherical caps), with the reasonable assumption of a smooth defect preparation via subtractive manufacturing. Some repair scenarios might involve more irregular or asymmetric geometries, which can introduce non-axisymmetric shock structures, spatially varying particle impact angles, and more complex particle-substrate interactions. For instance, highly asymmetric features may distort the bow shock structure, introduce local shock strengthening or weakening, and promote flow separation and recirculation zones. These effects lead to spatially heterogeneous gas velocity and temperature fields, as well as shadowing and local jet blockage, resulting in increased variability of particle velocity and temperature. Consequently, predictions obtained using smooth, axisymmetric defect geometries may not fully capture localized flow deviations. Nevertheless, in practical CS repair workflows, defects are rarely repaired in their as-damaged state, as sharp edges, surface irregularities, or inclusions can hinder deposition efficiency and promote shadowing effects. Therefore, some degree of premachining is commonly applied to obtain smoother geometries. Similarly, the current models predict the average thermo-kinetic properties of particles without explicitly considering their initial position within the gas jet. Since particle location can affect trajectory, residence time, and interaction with bow shocks, incorporating this spatial dependence could provide an additional layer of accuracy.

4. Conclusions and outlook

This study introduced an integrated CFD–AI–XAI framework for predicting in-flight thermo-kinetic properties of the Cold Spray process. The computational fluid dynamics (CFD) model demonstrated strong reliability, with particle velocity predictions showing mean percentage errors of 2.0% compared to analytical models and 3.6% compared to commercial KSS software, as well as good agreement with experimental velocimetry data. Temperature predictions achieved a mean percentage error of 6.1% relative to KSS software, highlighting the greater challenges in modeling gas-particle heat exchange.

The surrogate support vector regression models trained on CFD data provided accurate and computationally efficient predictions. Velocity was predicted with a coefficient of determination (R^2) of 0.982 and a mean average percentage error (MAPE) of 0.89%, while temperature predictions achieved an R^2 of 0.882 with a MAPE of 1.15%. These results confirm that the hybrid approach can capture the key thermo-kinetic trends of the process at a fraction of the computational cost of CFD simulations. Importantly, the integration of SHapley Additive exPlanations (SHAP) analysis ensured that predictive outcomes remained physically interpretable, with gas temperature, particle diameter, and gas pressure identified as the most influential factors.

Future developments might include expanding the training dataset with additional CFD simulations, particularly under varying defect geometries, which would improve model generalization. Different feedstock powders may also be studied to quantify how differences in particle size distributions, mass, and conductivity properties affect particle velocity and temperature. Upcoming work could also investigate multi-objective optimization, considering deposition efficiency, coating adhesion, and deposit quality.

In summary, the integration of CFD, artificial intelligence, and explainable artificial intelligence represents a robust and scalable pathway for advancing the understanding of Cold Spray processes and facilitating efficient process optimization. By combining physical fidelity, computational efficiency, and interpretability, the proposed framework facilitates real-time, data-driven control of coating and repair technologies.

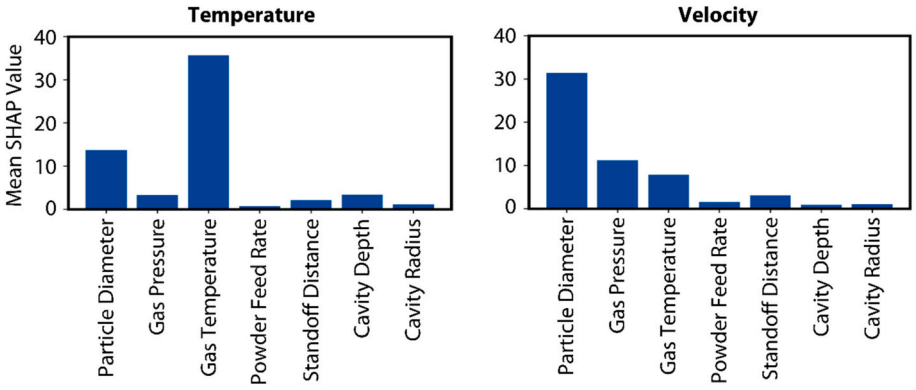

Fig. 12. Mean SHAP values of the SVR model prediction for particle temperatures and velocities.

CRediT authorship contribution statement

Roberta Falco: Writing – original draft, Validation, Software, Methodology, Data curation, Conceptualization. **Kirsten Bobzin:** Writing – review & editing, Supervision, Project administration, Funding acquisition. **Hendrik Heinemann:** Writing – review & editing, Supervision, Conceptualization. **Marvin Erck:** Writing – review & editing, Supervision, Project administration. **Kevin Jasutyn:** Writing – review & editing, Supervision. **Christopher Wasels:** Writing – original draft, Software, Methodology, Data curation. **Sara Bagherifard:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is supported by the ERC-CoG project ArcHIDep n. 101044228, and ERC-PoC grant CoRe n. 101188904, funded by the European Union. Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. This work is also funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612. We acknowledge the contributions of Dr. Ali Dokhanchi and Gabi Nassar for interesting discussions in the conceptualization phase of the present work.

Data availability

The source code used in this study is available by request on GitHub at https://github.com/ChristopherWasels/SHAP.

References

[1] Assadi H, Gärtner F, Stoltenhoff T, Kreye H. Bonding mechanism in cold gas spraying. Acta Mater 2003;51:4379–94. https://doi.org/10.1016/S1359-6454(03)00274-X.
[2] Papyrin A, Kosarev V, Klinkov S, Alkhimov A, Fomin VM. Cold spray technology. Elsevier; 2006.
[3] Champagne V, Helfritch D. The unique abilities of cold spray deposition. Int Mater Rev 2016;61:437–55. https://doi.org/10.1080/09506608.2016.1194948.
[4] Falco R, Bagherifard S. Cold spray additive manufacturing: a review of shape control challenges and solutions. J Therm Spray Tech 2025;34:1023–41. https://doi.org/10.1007/s11666-025-01970-0.
[5] Dang M, Guo Z, Ma T, Yang X, Su Y, Li W, et al. Microstructure evolution and mechanical properties of a linear friction welded TC4-DT titanium alloy joint. Mater Today Commun 2025;46:112684. https://doi.org/10.1016/j.mtcomm.2025.112684.
[6] Han B, Huang Y, Lv S, Wan L, Feng J, Fu G. AA7075 bit for repairing AA2219 keyhole by filling friction stir welding. Mater Des 2013;51:25–33. https://doi.org/10.1016/j.matdes.2013.03.089.
[7] Bagherifard S, Guagliano M. Fatigue performance of cold spray deposits: coating, repair and additive manufacturing cases. Int J Fatigue 2020;139:105744. https://doi.org/10.1016/j.ijfatigue.2020.105744.
[8] Lee M-W, Park J-J, Kim D-Y, Yoon SS, Kim H-Y, James SC, et al. Numerical studies on the effects of stagnation pressure and temperature on supersonic flow characteristics in cold spray applications. J Therm Spray Tech 2011;20:1085–97. https://doi.org/10.1007/s11666-011-9641-1.
[9] Yin S, Zhang M, Guo Z, Liao H, Wang X. Numerical investigations on the effect of total pressure and nozzle divergent length on the flow character and particle impact velocity in cold spraying. Surf Coat Technol 2013;232:290–7. https://doi.org/10.1016/j.surfcoat.2013.05.017.
[10] Park J-J, Lee M-W, Yoon SS, Kim H-Y, James SC, Heister SD, et al. Supersonic nozzle flow simulations for particle coating applications: effects of shockwaves, nozzle geometry, ambient pressure, and substrate location upon flow characteristics. J Therm Spray Tech 2011;20:514–22. https://doi.org/10.1007/s11666-010-9542-8.
[11] Tabbara H, Gu S, McCartney DG, Price TS, Shipway PH. Study on process optimization of cold gas spraying. J Therm Spray Tech 2011;20:608–20. https://doi.org/10.1007/s11666-010-9564-2.
[12] Yin S, Wang X, Li W, Xu B. Numerical study on the effect of substrate angle on particle impact velocity and Normal velocity component in cold gas dynamic spraying based on CFD. J Therm Spray Tech 2010;19:1155–62. https://doi.org/10.1007/s11666-010-9510-3.
[13] Gärtner F, Stoltenhoff T, Schmidt T, Kreye H. The cold spray process and its potential for industrial applications. J Therm Spray Tech 2006;15:223–32. https://doi.org/10.1361/105996306X108110.
[14] Ardeshiri Lordejani A, Romanenghi L, Pollastri A, Guagliano M, Bagherifard S. Deposit shape control for local repair and welding by cold spray. J Manuf Process 2024;112:45–59. https://doi.org/10.1016/j.jmapro.2024.01.023.
[15] Mehmood A, Zunaid M, Madan AK. Multi-objective optimization and performance improvement of CD nozzle design parameters for cold spray coating process using RSM. ANN and GA Materials Today Communications 2024;40:109446. https://doi.org/10.1016/j.mtcomm.2024.109446.
[16] Meng Y, Saito H, Bernard C, Ichikawa Y, Ogawa K. Optimal Design of a Cold Spray Nozzle for Inner Wall coating fabrication by combining CFD simulation and neural networks. J Therm Spray Tech 2024;33:3–16. https://doi.org/10.1007/s11666-024-01716-4.
[17] Falco R, Jalayer M, Bagherifard S. Enhanced geometrical control in cold spray additive manufacturing through deep neural network predictive models. Virtual Phys Prototyp 2025;20:e2472388. https://doi.org/10.1080/17452759.2025.2472388.
[18] Hamrani A, Medarametla A, John D, Agarwal A. Machine-learning-driven optimization of cold spray process parameters: robust inverse analysis for higher deposition efficiency. Coatings 2025;15:12. https://doi.org/10.3390/coatings15010012.
[19] Sharma D, Boruah D, Bakir AA, Ameen A, Paul S. Machine learning-based predictions of porosity during cold spray deposition of high entropy alloy coatings. Coatings 2024;14:404. https://doi.org/10.3390/coatings14040404.
[20] Barredo Arrieta A, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, et al. Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inf Fusion 2020;58:82–115. https://doi.org/10.1016/j.inffus.2019.12.012.
[21] Bobzin K, Heinemann H, Erck M, Nassar G. Reshaping thermal spraying: explainable artificial intelligence meets plasma spraying, ASM. International 2025:237–44. https://doi.org/10.31399/asm.cp.itsc2025p0237.
[22] Bobzin K, Wietheger W, Heinemann H, Dokhanchi SR, Rom M, Visconti G. Prediction of particle properties in plasma spraying based on machine learning. J Therm Spray Tech 2021;30:1751–64. https://doi.org/10.1007/s11666-021-01239-2.
[23] Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. In: Advances in neural information processing systems. vol. 30. Curran Associates, Inc.; 2017.
[24] Anderson J. Modern compressible flow. New York, NY: McGraw-Hill Education; 2021.
[25] Dykhuizen RC, Smith MF. Gas dynamic principles of cold spray. J Therm Spray Tech 1998;7:205–12. https://doi.org/10.1361/105996398770350945.
[26] Du Plessis JP, Masliyah JH. Mathematical modelling of flow through consolidated isotropic porous media. Transp Porous Media 1988;3:145–61. https://doi.org/10.1007/BF00820342.
[27] Leitz K-H, O'Sullivan M, Plankensteiner A, Kestler H, Sigl LS. OpenFOAM modeling of particle heating and acceleration in cold spraying. J Therm Spray Tech 2018;27:135–44. https://doi.org/10.1007/s11666-017-0644-4.
[28] William E. Ranz and W. Evaporation From Drops I And II: Robert Marshall. Ranz Marshall; 2024.
[29] Chase MW. NIST-JANAF thermochemical tables. 4th ed. American Institute of Physics; 1998.
[30] Yin S, Meyer M, Li W, Liao H, Lupoi R. Gas flow, particle acceleration, and heat transfer in cold spray: a review. J Therm Spray Tech 2016;25:874–96. https://doi.org/10.1007/s11666-016-0406-8.
[31] Mckay MD, Beckman RJ, Conover WJ. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 2000;42:55–61. https://doi.org/10.1080/00401706.2000.10485979.
[32] Shields MD, Zhang J. The generalization of Latin hypercube sampling. Reliab Eng Syst Saf 2016;148:96–108. https://doi.org/10.1016/j.ress.2015.12.002.
[33] KSS kinetic spray solutions – simulation in gas dynamic cold spraying. https://www.kinetic-spray-solutions.com/. [Accessed 2 July 2025].
[34] Schmidt T, Assadi H, Gärtner F, Richter H, Stoltenhoff T, Kreye H, et al. From particle acceleration to impact and bonding in cold spraying. J Therm Spray Tech 2009;18:794–808. https://doi.org/10.1007/s11666-009-9357-7.
[35] Stoltenhoff T, Kreye H, Richter HJ. An analysis of the cold spray process and its coatings. J Therm Spray Tech 2002;11:542–50. https://doi.org/10.1361/105996302770348682.
[36] Vapnik VN. The support vector method. In: Gerstner W, Germond A, Hasler M, Nicoud J-D, editors. Artificial neural networks — ICANN'97. Berlin, Heidelberg: Springer; 1997. p. 261–71. https://doi.org/10.1007/BFb0020166.
[37] Smola AJ, Schölkopf B. A tutorial on support vector regression. Stat Comput 2004;14:199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88.

[38] Nasrabadi NM. Pattern recognition and machine learning n.d.
[39] Shapley LS. A value for n-person games. In: Roth AE, editor. The Shapley value: Essays in honor of Lloyd S. Shapley. Cambridge: Cambridge University Press; 1988. p. 31–40. https://doi.org/10.1017/CBO9780511528446.003.
[40] Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. In: Proceedings of the 31st international conference on neural information processing systems. Red Hook, NY, USA: Curran Associates Inc; 2017. p. 4768–77.