Reconstruction of particle image velocimetry data using flow-based features and validation index: a machine learning approach

Ghasem Akbari1,∗ and Nader Montazerin2

1 Department of Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
2 Department of Mechanical Engineering, Amirkabir University of Technology, Tehran, Iran

E-mail: g.akbari@aut.ac.ir

Received 4 July 2021, revised 2 October 2021
Accepted for publication 5 October 2021
Published 22 October 2021

Abstract
Reconstruction of flow field from real sparse data by a physics-oriented approach is a current challenge for fluid scientists in the AI community. The problem includes feature recognition and implementation of AI algorithms that link data to a physical feature space in order to produce reconstructed data. The present article applies machine learning approach to study contribution of different flow-based features with practical fluid mechanics applications for reconstruction of the missing data of turbomachinery particle image velocimetry measurements. Support vector regression (SVR) and multi-layer perceptron (MLP) are selected as two robust regressors capable of modelling non-linear fluid flow phenomena. The proposed flow-based features are optimally scaled and filtered to extract the best configuration. In addition to conventional data-based validation of the regressors, a metric is proposed that reflects mass conservation law as an important requirement for a physical flow reproduction. For a velocity field including 25% of clustered missing data, the reconstruction accuracy achieved by SVR in terms of $R^2$-score is as high as 0.993 for the in-plane velocity vectors in comparison with that obtained by MLP which is up to 0.981. In terms of mass conservation metric, the SVR model by $R^2$-score up to 0.96 is considerably more accurate than the MLP estimator. For extremely sparse data with a gappiness of 75%, vector and contour plots from SVR and MLP were consistent with those of the original field. Comparison of the ML models outcome with the reconstruction made by six additional spatial interpolation methods indicates considerably higher performance of the SVR in terms of either the velocity-based or mass conservation based metrics.

Keywords: machine learning (ML), flow-based feature, velocity reconstruction, support vector regression (SVR), multi-layer perceptron (MLP), particle image velocimetry (PIV), flow-based validation index

(Some figures may appear in colour only in the online journal)

∗ Author to whom any correspondence should be addressed.
1. Introduction

Technological race towards building reliable velocity measuring instrumentation with high temporal and spatial resolution is moving fast. Particle image velocimetry (PIV) and particle tracking velocimetry (PTV) are two expensive methods that are prone to gappy data and require careful post processing [1]. This needs reliable schemes that recompense the use of affordable low resolution equipment, with the calculation of high frequency/wave number flow structures and reconstruction of the missing data.

Velocity reconstruction with physical methods results in high resolution and is powerful against noise or data sparsity. Use of the vorticity transport equation is one such methodology that works independent of boundary conditions [2]. An ‘all-in-one method’ was developed for validation of PIV raw data, correction of spurious and missing data and smoothing of the outcome field [3]. The penalized least square method was also used to automatically classify the flow outliers into scattered and clustered ones [4]. These methods are computationally expensive and are only applicable where numerical pre-estimates of the velocity field are achievable or at least where boundary and initial conditions are known a priori [5, 6].

Statistical methods identify and subsequently eliminate individual vectors that depart from the ensemble of the recorded velocity flow field and eventually substitute them with the corrected vectors [1, 7–10]. In just a few examples of such methods, Nogueira et al proposed a linear approach based on local Taylor expansions to filter and correct false vectors in PIV data [11]. This was robust and resulted in straightforward optimizations with limited accuracy. The PIV missing data was corrected in another study through element-by-element multiplication of correlation matrix calculated from adjacent PIV nodes [12]. This approach resulted in bias error reduction and subpixel accuracy enhancement. Bootstrapping is another approach that independently generates statistics for each velocity component and is capable of both outlier detection and correction of flow field [13].

Application of proper orthogonal decomposition (POD) to extract the DNS- and PIV-based modes as well as utilization of Kriging interpolation to smooth the obtained POD modes has proven to enhance reconstruction accuracy but with increased computational cost [14]. Everson and Sirovich [15] proposed a physical POD based approach combined with the least squares scheme for flow field image reconstruction from gappy data. Venturi and Karniadakis [16] developed an extended version termed gappy POD (GPOD). It was robust and effective, while its higher computational cost in comparison with that of Everson and Sirovich approach was a drawback. The limitation of using a constant number of POD modes in traditional gappy POD methods was overcome by implementing a local adaptive criterion for detecting an optimized count of modes [17].

On outlier detection and reconstruction, Wang et al [18] proposed a more efficient but slightly slower than conventional POD approach. This method iteratively identified and replaced clustered outliers using POD and dynamically reproduced the target velocity field. A successful algorithm for detection of outliers for single PIV frames and their velocity field reconstruction without decreasing the estimation accuracy was proposed based on a non-iterative alternative POD scheme [19]. Another gappy POD method, applied to the PIV data of high-pressure gas turbine combustor flow with low signal-to-noise ratio (SNR), adaptively decided whether a local missing data is updated in each iteration [20]. This method resulted in lower reconstruction error than the previous GPOD approaches as well as the Kriging interpolation. Application of a POD-based fusion approach was successful in recovering the missing data, especially when the flow field had dominating flow structures that were closely correlated in space [21]. Such approaches do not consider the physical characteristics of the flow and normally smooth velocity gradients or lower spatial resolutions.

Nowadays, machine learning (ML) algorithms are extensively utilized for a diverse range of fluid flow problems. Powerful capability of ML in forming a generalized non-linear relationship between input space and target flow quantities makes it a plausible contender for handling complicated flow data [22]. Among different applications of ML algorithms in PIV-based flow analysis or related disciplines, four items are highlighted in the following:

(a) Handling raw images to improve their defects for further post-processing: Implementation of convolutional neural network (CNN) to Remove spurious particle images and efficiently regenerate them with the same resolution works for either low- or high-particle seeding reconstructions [23]. CNN was also successfully applied by Gao et al [24] to refine the particle reconstructions from very coarse initial guess and any classic algebraic PIV processing technique.

(b) Implementation of ML approach to extract the velocity field from raw acquired PIV images: Rabault et al [25] were the first to apply CNN and fully connected neural network with PIV images to compute the velocity field. In comparison with the conventional cross-correlation methods, a better near-edge performance and higher spatial resolution was reported. Cai et al [26, 27] also used a deep learning estimator in terms of a CNN to extract the velocity field from a pair of PIV images. They showed similar accuracy as other state of the art methods and high efficiency toward a real-time estimation for both artificial and experimental images. Satisfactory accuracy and computational efficiency were also obtained from a shallow neural network model for PTV application [28]. Barway et al [29] used CNN to a combination of OH-PLIF (planar laser-induced fluorescence of hydroxyl radical) and PIV to build a velocity field in a strong swirling combustor. They decoded the three-dimensional PIV fields from input OH-PLIF images, as a practical approach when simultaneous measurements were limited. Tombal et al [30] applied both artificial neural network and support vector machine (SVM) to PIV data of particle transport in multiphase flow. They showed that SVM as a classifier was the best for direction estimation and support vector regression (SVR) was the best for velocity approximation.
Utilization of ML algorithms to enhance temporal or spatial resolution of PIV data: Time-resolved flow reconstruction is another recent outcome of ML. Implementation of long short-term memory-based networks and bidirectional recurrent neural networks, and their combination with POD confirmed the potential of this approach for accurate time-series reconstruction of low-frequency PIV data [31, 32]. Giannopoulos and Aider [33] used neural network and time-resolved PIV data and successfully predicted temporal POD coefficients of the flow using visual sensors from local velocity measurements. They found a shallow neural network as a sufficient tool to obtain reasonable accuracy with low computational time. Liu et al [34] utilized two deep learning models on DNS data of channel flow to develop spatial resolution enhancement. They obtained reproduction accuracy in capturing the anisotropic characteristics in the wall regions of flow field. The data resolution enhancement was also achieved after applying generative adversarial network to the PIV data of the complicated wake flow behind two side-by-side cylinders [35].

Reproduction of velocity data at the gappy regions by ML models: Implementation of neural network on three artificially-generated flow data was used to filter the inherent noise and reconstruct the flow field at the gappy regions [36]. There was agreement between the predicted output and the theoretical target data for training a network based on either exact or noisy data. Utilization of neural network with radial basis function (RBF) was also reported as a robust interpolating method for reconstructing noisy data from PIV of a swirling flow [37]. The RBF interpolation scheme was also utilized for PTV data and its outcome was superior to those of standard Gaussian weighting and Taylor expansion interpolation methods [38]. Extreme learning machine auto-encoders was another methodology applied on sparse flow data and found to be as a competitive reconstruction approach compared with the POD-based methods [39].

The above algorithms commonly utilize the spatial coordinate as the input feature of the ML regressors without implementation of flow-based quantities. Substantial capability of ML algorithms in reproduction of sparse flow fields, shows the potential for moving from a data-based approach towards a physics-based approach. The present study aims to use these potentials for proposing a novel approach in which flow-based features with practical fluid mechanics characteristics are considered in addition to the conventional coordinate-based features. The objective is reconstruction of the velocity vectors in any instantaneous PIV-measured turbomachinery flow field at the gappy regions in which spurious data are detected and removed. The path is to identify features and metrics that are native to fluid mechanics problems and facilitate ML algorithms. Multi-layer perceptron (MLP) and SVR are two ML models adopted due to their suitable performance in handling highly non-linear flow attributes in the complicated turbomachinery field. In addition to conventional evaluation of ML models based on the velocity data, a metric is proposed that relies on mass conservation principle as an important requisite for a physical reconstructed field.

The article is organized as follows. Section 2 introduces the overall procedure, including PIV data acquisition, data validation procedure, ML pre-processing steps, training of data by SVR and MLP, and hyper-parameter optimization. Model evaluation metrics are introduced in section 3. Results and discussion follow in section 4 presenting various aspects of flow field reconstruction and validation.

2. PIV data acquisition, process and reconstruction methodology

There are eight tasks to acquire initial experimental data as raw PIV images, process them into validated velocity vectors and reconstruct them to predict the missing data (as depicted in figure 1). In the following, these steps are discussed in more details.

2.1. Experimental setup, data acquisition and raw-image processing

Stereoscopic PIV (SPIV) is utilized to measure three instantaneous velocity components at various points of a planar Cartesian grid. The data is acquired from a constructed forward-blade centrifugal fan where images are recorded at an encoded position of the rotor. The experimental setup and instrumentation include the following components (as illustrated in figure 1):

- A 43-blade centrifugal fan constructed from galvanized plate (table 1)
- A test setup and an outlet duct compatible with ISO 5801 [40]
- A volute made of transparent Plexiglas to provide optical access to the test section
- A double-cavity Quantel Brilliant Nd-YAG laser equipped with an optical guide system that delivers the laser sheet to the test section
- Two FlowSense® 1600 × 1186 pixel double-frame CCD cameras
- SAFEX F2010 plus fog generator
- Dantec FlowMap® system for synchronization of laser, camera and encoder actions.

Velocity measurement is performed at a plane perpendicular to the rotor axis at the upper region of the rotor with \( z/B = 0.6 \) where \( z \) is the axial coordinate, measured from the volute inlet, and \( B \) is the volute width (figure 2(a)). Previous experimental studies on this centrifugal turbomachine indicate that the selected region involves the most intensive mutual interactions between jet-wake-volute flow structures [41, 42]. This complicated flow field would be a desirable candidate to evaluate performance of ML algorithms in correct reconstruction of the field at target regions. The rectangular measurement area is divided to four overlapping regions, termed as field of view (FOV), in order to both measure a fairly extensive region
and resolve sufficiently detailed flow information (figure 2(b)). The size of each FOV is 100 $\times$ 74 mm$^2$ with velocity vector spacing ($\delta$) in both $x$ and $y$ directions to be 1 mm. The coordinate origin for each FOV is located at its lower left corner.

FlowManager® software [44] is used to extract the velocity components from two pairs of double-frame images captured simultaneously from different views. Multi-stage adaptive cross-correlation is utilized to obtain the raw 2D vector map for each separate image. More specifically, adaptive cross-correlation process is initiated over each 256 $\times$ 256 pixel interrogation window and after two stages of adaptive correlation, the vector maps are extracted for 64 $\times$ 64 pixel areas. A 50% overlap is considered between neighbouring interrogation areas in either horizontal or vertical directions in order to manage the unfavourable effect of loss of pairs. The acquired data consists of 2000 instantaneous snapshots for each FOV.
Table 1. Geometrical characteristics of the centrifugal fan [43].

| Region  | Parameter         | Value  | Parameter         | Value  |
|---------|-------------------|--------|-------------------|--------|
| Rotor   | Inner diameter, $D_1$ | 285 mm | Outer diameter, $D_2$ | 350 mm |
|         | Number of blades, $N$ | 43     | Rotor width, $b$   | 165 mm |
|         | Blade entrance angle, $\beta_1$ | 90°    | Blade exit angle, $\beta_2$ | 155°   |
| Volute  | Volute width, $B$  | 200 mm | Outlet height, $h$ | 200 mm |
|         | Scroll spread angle, $\alpha$ | 5°     | Cut-off angle, $\gamma$ | 60°    |
|         | Cut-off radius, $r_0$ | 210 mm |Rotor cut-off gap, $S_z$ | 35 mm  |
| Inlet   | Inner diameter, $d_1$ | 295 mm |Outer diameter, $d_2$ | 305 mm |
|         | Curvature radius, $r_1$ | 15 mm  |                    |        |

Figure 2. (a) Geometrical characteristics of the centrifugal fan; (b) location and dimension of the four fields of view at the rotor exit region. Reproduced with permission from [43].

2.2. PIV data validation and 3D vector processing

The PIV vector maps after adaptive cross-correlation may contain some unreasonable and incorrect vectors, called ‘spurious vectors’ or ‘outliers’. They could appear scattered over an extended area or clustered at particular locations of flow field. The reasons for existing spurious vectors could be inhomogeneous dispersion of seeding particles, low intensity of laser illumination, optical noise and poor quality of a captured image. Prior to any velocity field analysis, it is important to detect and substitute spurious vectors, either scattered or clustered.

Data validation procedure, including velocity-range validation and peak-height validation are employed to eliminate spurious velocity vectors [45]. In the velocity-range validation, the reasonable velocity range for each FOV is specified based on the previous literature and experimental studies [43]. The spurious vectors are identified as the vectors with velocity outside the acceptable range. The peak-height validation method evaluates the quality of data signal by computing the ratio of the highest and the second highest correlation peaks, called as the SNR [12, 46]. Sufficiently large value of SNR in the correlation procedure is the necessity of achieving reliable
displacement vectors from raw PIV images. The minimum threshold of SNR recommended by Keane and Adrian [46] is 1.2 which is used in the peak-height validation of the present study. Such minimum threshold accepts 82%–90% of samples in FOVs 1 and 2 and more than 98% of samples in FOVs 3 and 4 [41].

Two concurrent 2D vector maps are then combined based on the calibration mapping model in order to achieve a unified planar vector map including three velocity components for each instantaneous snapshot. This mapping model is obtained using FlowManagements® software after applying a third-order XYZ-polynomial method on the captured calibration images [44]. The resulted three-component vector map includes regions with missing data from the validation procedure that could be imputed by ML approach.

The overall uncertainty due to displacement estimation, particle lag effect and stereoscopic imaging was assessed to be 1.5% for the in-plane and 4% for the out-of-plane velocity components [43].

2.3. Flow-based feature extraction and scaling

Data reconstruction in this paper pertains to different instantaneous snapshots, each contains three velocity components in the Cartesian grid of 98 × 73 with some missing data, either scattered or clustered. Each velocity component is the target of ML model for reconstruction, and appropriate input features should be provided for a robust training and prediction.

Because of the dependence of flow characteristics to geometry and position, the most basic input feature would be the spatial coordinate. It has been the conventional feature, utilized in the literature for reconstruction purpose [36–38]. The planar (x and y) coordinate of each velocity sample based on the regular PIV grid is consequently selected as the first category of input features, which is termed ‘type A’ feature hereafter.

Implementation of suitable flow-based features with fluid mechanics significance would enhance the velocity reconstruction performance due to their inherent correlations with the flow field. The present study proposes two types of features in this regard: ‘type B’ are features characterizing by the measured velocity at the neighboring positions and ‘type C’ are features relying on spatial gradient of velocity field at the target coordinate. Before describing these feature types, it is important to notice that any extracted feature should be non-zero in order to be used in a ML regressor. However, for clustered gappy data, existence of several null data in the neighbouring regions is quite possible. In order to impute such null features, it is proposed to apply filtering operation on both of the above-mentioned flow-based features over a customized rectangular domain.

‘Type B’ features are calculated after filtering the original PIV velocity field with different filter scales. In this way, the velocity data at the proximity of each target point contributes to the calculated feature. It takes advantage of similarity between neighbouring velocity vector, particularly where the target point with null data is surrounded by non-zero velocity vectors. The general filtering operation on each instantaneous velocity component over a spatial domain D is given as [47]:

$$\pi_l(i_1, i_2) = \int_D G(x_1-x_i, x_2-x_i) \mu_i(x_i, x_i) dx_1 \, dx_2$$

(1)

where the overbar denotes the filtering operation, \( x_1 \) and \( x_2 \) are spatial coordinates in x and y directions and G is the filter kernel. \( i \) indicates the velocity components which are 1, 2 and 3 for x, y and z directions, respectively. The top-hat (box) and Gaussian kernels have been frequently utilized in fluid dynamics scope [48] which are selected in our study as well. Because of discrete values of PIV velocity, the continuous filtering is not applicable and a discontinuous form should be utilized. The discrete kernels applied for the top-hat and Gaussian filters are [47]:

$$G_{\text{top–hat}}(i_1, i_2) = \begin{cases} C_1 & \text{if } |x_1| < \Delta_x \text{ and } |x_2| < \Delta_y, \\ 0 & \text{otherwise} \end{cases}$$

(2)

$$G_{\text{Gauss}}(i_1, i_2) = C_2 \exp \left[ \frac{-6(x_1^2 + x_2^2)}{\Delta_x \Delta_y} \right]$$

(3)

\( \Delta_x \) and \( \Delta_y \) are the filter scales in x and y directions, respectively. The coefficients \( C_1 \) and \( C_2 \) are normalization factors which ensure that the integral of the filter kernel over the filtering region equals unity (in a discrete manner). The null velocity at gappy locations should be excluded from filtering operation. Consequently, \( C_1 \) and \( C_2 \) are dependent to the count and position of non-zero velocities.

The ‘type C’ features are defined to be the filtered deformation-based quantities (filtered strain rate tensor and filtered vorticity vector) as frequently-used attributes in fluid mechanics. The filtered strain rate tensor (S) as the symmetric part of filtered deformation tensor is defined as [49]:

$$\mathbf{S} = \frac{1}{2} \left( \nabla \mathbf{V} + \nabla \mathbf{V}^T \right)$$

(4)

where \( \nabla \) is the filtered velocity vector and the superscript T notifies the matrix transpose operation. Furthermore, the filtered vorticity vector (\( \mathbf{\omega} \)) as the anti-symmetric part of filtered deformation tensor is calculated as curl of the filtered velocity vector:

$$\mathbf{\omega} = \nabla \times \mathbf{V}.$$  

(5)

Because of planar nature of PIV measurements, the in-plane components of the filtered strain rate tensor (equation (4)) and the out-of-plane component of the vorticity vector (equation (5)) are calculable.

Since the range of values for three proposed feature types are different, prior to importing them to the ML regressors, it is necessary to scale the features by standard methods. Four conventional approaches are applied in the present study for this purpose: (a) ‘normalizer’ method that scales each individual sample to the unit norm; (b) ‘min–max scaler’ that transforms
each feature by scaling it to a customized range; (c) ‘standard scaler’ that standardizes each feature vector to a distribution with zero mean and unit variance; (d) ‘robust scaler’ that scales the feature vectors based on customized quantile range and is robust to outliers.

2.4. Artificial gap generation

In order to evaluate the ML model accuracy, it is necessary to add some artificial gaps to provide unseen data for the trained model. The gap percentage, GP, is defined as the ratio of the number of gappy data in the active (non-masked) region of each snapshot to the overall number of points in that region. The initial value of gap percentage depends on the extent of spurious vectors. A gappiness threshold (in percent), GT, is the target situation in which the extent of gappy data (including the artificial gaps) reaches the desired quantity. The higher the assigned gappiness threshold, the more challenging accurate prediction of velocity data in the gappy regions is. Utilizing a gap generator, we add numerous rectangular masks with random dimensions in specified range and apply them in randomly adopted locations through the FOV, until reaching to the customized gappiness threshold (GP ≥ GT). Considering the space pertaining to artificially generated gappy data as $S_G$, the masked velocity components ($u^\text{mask}_i$, $i = 1, 2, 3$) are given as:

$$u^\text{mask}_i = \begin{cases} 0 & \text{if } R \in S_G \\ u^\text{org}_i & \text{otherwise} \end{cases} \quad (6)$$

where $u^\text{org}_i$, $i = 1, 2, 3$ is the original velocity components (including the detected spurious velocity vectors as null data) and $R$ is the two-dimensional spatial vector (coordinate) with respect to the FOV origin. For each instantaneous snapshot, a portion of flow data from the space $S_G$ with given velocity components is considered as ‘test data’ to evaluate the ML model accuracy and the remainder is randomly split into the training and validation sets. The training data is utilized to calculate the weight coefficients in each ML model while the validation portion is used to evaluate the model accuracy and tuning the model hyper-parameters.

2.5. Training the machine learning models

SVR and feed-forward MLP neural network are two conventional ML approaches utilized for modelling and reconstruction of the velocity field. There are several merits for selecting these two models as estimators. Implementation of numerous layers in conjunction with applying non-linear activation functions in MLP and applying the SVR kernel with non-linear characteristics introduce them as two powerful tools in estimation of the complicated fluid flow phenomena for moderate amount of data [22].Flexibility of MLP in learning process regardless of network structure and robustness of SVR against overfitting, particularly in higher dimensions of feature space, and its reasonable computational cost are the other supportive reasons for such selection.

Reconstruction of three velocity components by either SVR or MLP is performed independently in separate regression processes. In fact, we experienced a lower accuracy when three velocity components were reconstructed simultaneously in a single SVR or MLP regression. This could be due to different uncertainties of SPIV measurements for various velocity components (particularly the higher uncertainty of the out-of-plane velocity component) that adversely affects regression accuracies for each individual component. Furthermore, the proposed model for either of SVR or MLP regressors is trained based on each individual (instantaneous) snapshot. This is due to low frequency of PIV data acquisition in the present study that significantly reduces temporal correlation between successive snapshots. Instead, the ML model metrics obtained for various instantaneous snapshots are averaged in order to stabilize the results for our analysis. The technical details of the SVR and MLP models are described as follows.

2.5.1. Support vector regression

SVM originated by Vapnik et al [50] is one of the most popular algorithms in supervised learning, either in regression or classification applications. SVM transforms the input data into a high dimensional feature space by utilizing a nonlinear map and simply performs a linear regression in that feature space. In other words, for a given set of training vectors $S = \{(x_i, y_i) : i = 1, \ldots, N\}$, a linear model is fitted as $\hat{y}(x) = w^T \phi(x) + b$ to minimize a cost function defined as [50]:

$$R = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{N} L(y_i, W^T \phi(x)) \quad (7)$$

where the smoothness of the model is controlled by the coefficient vector $w$, and $b$ is a bias parameter. $\phi(x)$ is a function projecting the input space into the feature space, $x_i$ denotes an input feature vector (coordinate and flow-based features) and $y_i$ indicates its corresponding target value (each individual PIV velocity component). $C$ is a positive coefficient determining the degree of penalized loss when training error occurs and the so-called Vapnik’s $\varepsilon$-insensitive loss function, $L$, for the customized radius of the insensitive tube $\varepsilon$ is considered as follows [51]:

$$L = \begin{cases} |y_i - (w \cdot x_i + b)| - \varepsilon & \text{for } |y_i - (w \cdot x_i + b)| > \varepsilon \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

The following optimization problem should be solved to train the above-mentioned model:

$$\min \left( \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{N} (\xi_i + \xi_i^*) \right) \quad (9)$$

subject to the following constraints for $i = 1, \ldots, N$

$$y_i - (w^T \phi(x_i) + b) \leq \varepsilon + \xi_i$$
$$w^T \phi(x_i) + b - y_i \leq \varepsilon + \xi_i^*$$
$$\xi_i, \xi_i^* \geq 0 \quad (10)$$
where \( \xi_i \) and \( \xi^*_i \) represent the slack variables corresponding to upper and lower training errors, respectively for an error tolerance \( \varepsilon \). Minimization of the Lagrange function, made from the objective function and the problem constraints, and adopting the dual set of Lagrange multipliers, \( \alpha_i \) and \( \alpha^*_i \), leads to the following dual form of the nonlinear SVM regression problem [51]:

\[
\min \left( \frac{1}{2} \sum_{i,j=1}^{N} (\alpha_i - \alpha^*_i) (\alpha_j - \alpha^*_j) K(x_i, x_j) + \varepsilon \sum_{i=1}^{N} (\alpha_i + \alpha^*_i) \right) + \sum_{i=1}^{N} y_i (\alpha_i - \alpha^*_i)
\]

subject to the constraints:

\[
\sum_{i=1}^{N} (\alpha_i - \alpha^*_i) = 0 \quad \text{and} \quad \alpha_i, \alpha^*_i \in [0, C].
\]

The function \( K(x_i, x_j) \) is a kernel matrix equivalent to the dot product of \( \phi(x_i) \) and \( \phi(x_j) \) which is evaluated by a kernel function. The selected kernel in the present study is the frequently-used Gaussian (RBF) kernel due to its flexibility in handling the non-linear patterns. This kernel function is given as:

\[
K(x_i, x_j) = \exp \left( -\gamma \| x_i - x_j \|^2 \right)
\]

where \( \gamma \) is a parameter that can be interpreted as the inverse of the radius of influence of samples selected by the model as support vectors. The model output is finally expressed in terms of Lagrange multipliers:

\[
\hat{y}(x) = \sum_{i=1}^{N} (\alpha_i - \alpha^*_i) K(x_i, x) + b.
\]

Accordingly, the SVM regression model can be obtained after training a quadratic problem for specified hyper-parameters, namely \( C, \varepsilon \) and \( \gamma \).

2.5.2. Feed-forward multi-layer perceptron. The second ML algorithm used to construct the PIV velocity field is the feed-forward MLP neural network coupled with back-propagation approach. The input layer consists of the features selected for learning the network and the output layer includes each velocity component. In order to increase the number of hidden layers and generate a more complicated neural network, it is necessary to utilize nonlinear activation functions for each neuron. Otherwise the relation between the input and output data is achievable via a single-layer perceptron. Figure 3 illustrates the architecture of a two-layer MLP neural network for reconstruction of each individual velocity component \((u_p : p = 1, 2, 3)\) based on the input feature vector, \( \mathbf{x}^p \) (including coordinate and flow-based features extracted from the \( p \)th velocity component field). The number of neurons in the first and second hidden layers is \( N_1 \) and \( N_2 \), respectively. The output signal of the \( j \)th neuron located at the \( k \)th layer is obtained as:

\[
y^j_k = \varphi(y^j_k) \quad j = 1, \ldots, N_k, \quad k = 1, \ldots, M
\]

where \( N_k \) is the number of neurons in the \( k \)th layer, \( M \) is the number of hidden layers (two layers for the architecture shown in figure 3) and \( y^j_k \) is the induced local value produced at the input of the activation function \( (\varphi(\cdot)) \). This input value is calculated based on a linear combination of all preceding interconnected neuron weights \((w^j_k)^p\) and their corresponding outputs \((y^i_{k-1})\) expressed as:

\[
y^j_k = \sum_{i=0}^{N_{k-1}} w^j_k y^i_{k-1}, \quad j = 1, \ldots, N_k, \quad k = 1, \ldots, M.
\]
The rectifier linear unit (ReLU) function is implemented as the activation function given as:

\[ \varphi(v) = \max(0, v) \]  

(17)

The ReLU function avoids and rectifies the vanishing gradient problem in the gradient descent optimization of weights. Moreover, the ReLU is less computationally expensive than other well-known activation functions such as sigmoid and hyperbolic tangent because it involves simpler mathematical operations.

The learning algorithm includes forward and backward passes which iterate until satisfaction of the convergence criterion in which the error signal tends to zero and the MLP output tends to the actual PIV data for the training data set. The updated synaptic weights in the last iteration are utilized for reconstruction of the velocity components (the network output) after one forward pass throughout the MLP.

2.6. Optimization of hyper-parameters via k-fold cross-validation

Optimal values of model hyper-parameters should avoid overfitting and lead to suitable generalization performance for the model. Grid search and random search are two frequently used schemes to this end. The random search approach was proved empirically and theoretically more efficient for hyper-parameter optimization than the grid search scheme [52] and is consequently applied in the present study.

Based on the random search scheme, a wide range of hyper-parameters are considered and a series of ML models are trained. A k-fold cross validation procedure is employed to evaluate the model performance and in order to avoid occurrence of overfitting. After splitting training data set into k subsets of equal size, each subset is used for the model evaluation by feeding the remaining k−1 subsets as the input data to the model for training purpose. This is iterated until each of k subsets was used for validation. The model is then evaluated based on a suitable error metric for each of k validation subsets and the average of calculated errors is considered as the representative error for the hyper-parameter combination used. The minimum error among all different combinations is selected after comparing the averaged error metric calculated for various randomly adopted hyper-parameters.

3. Model evaluation and error metrics

Several error metrics could be applied to evaluate the model accuracy either in the validation or test stages. Since the PIV data contains numerous outliers, the median absolute error (MedAE) of each velocity component is the first metric selected for model evaluation because it is less sensitive to outliers in comparison with conventional metrics such as mean absolute error (MAE) and mean squared error. Furthermore, because of different scales of the three velocity components (as the regressor target), the median of all normalized absolute errors between target and predicted values is regarded as MedNAE, and is defined as follows:

\[ \text{MedNAE} = \text{median} \left( \frac{|y_1 - \bar{y}_1|}{|\bar{y}_1|}, \frac{|y_2 - \bar{y}_2|}{|\bar{y}_2|}, \ldots, \frac{|y_N - \bar{y}_N|}{|\bar{y}_N|} \right) \]  

(18)

where \( y \) and \( \bar{y} \) denote the target and predicted values of the velocity components, respectively and \( N \) is the number of data samples. The closer to zero is this error metric, the more accurate the predictor performs.

The second frequently-used metric is the coefficient of determination, also termed as \( R^2 \)-score. It is expressed as:

\[ R^2 = 1 - \frac{\sum_{i=1}^{N_s} (y_i - \bar{y}_i)^2}{\sum_{i=1}^{N_s} (y_i - \langle y \rangle)^2} \]  

(19)

where \( \langle y \rangle \) is the average of target values, \( \bar{y} = \left( \frac{\sum_{i=1}^{N_s} y_i}{N_s} \right) \). This error metric indicates how good the model fits and how good it expresses the amount of variability in the target quantity that is accounted for by the predictor variables [53]. Best possible score of \( R^2 \) is unity, while the value of zero pertains to the case in which the model always predicts the same value, regardless of the input feature values (an uncorrelated model).

In addition to the above mentioned velocity-based metrics, as a third metric we propose to evaluate the mass conservation for the original and reconstructed velocity fields as a fundamental fluid mechanics requirement. For an incompressible flow with continuous velocity field, the mass conservation equation is given as follows:

\[ \nabla \cdot \mathbf{U} = 0. \]  

(20)

In the case of planar PIV data, the in-plane velocity gradients are calculable by finite difference approach and based on the neighboring velocity components. Accordingly, the expanded form of equation (20) could be written as follows, in which the right-hand-side terms are directly calculable from the planar velocity data:

\[ \left( \frac{\partial u_1}{\partial x} \right)_{(i,j)} = - \left( \frac{u_1^{(i+1,j)} - u_1^{(i-1,j)}}{2\delta x} + \frac{u_2^{(i,j+1)} - u_2^{(i,j-1)}}{2\delta y} \right). \]  

(21)

The subscripts 1, 2 and 3 in equation (21) denote the \( x \), \( y \) and \( z \) coordinates, respectively. The superscripts \( i \) and \( j \) notify the \( x \) and \( y \) positions of the target point in the discrete PIV map. The left-hand-side term in equation (21) could be considered as a metric that evaluates the out-of-plane mass flux in a conservative way. This is termed as mass conservation factor (MC) hereafter. Since the final objective is to reconstruct the velocity field in a consistent manner to the measured PIV field, a higher ML accuracy occurs when MC, calculated from the reconstructed velocity field is as close as possible to that of the original PIV data. It should be noticed that mass conservation factor is defined based on a three-dimensional flow field and it is calculable from the in-plane velocity components. In fact,
based on equation (21), the mass conservation factor would be zero for a truly two-dimensional flow field, while it could take any value for a 3D flow field (similar to that experienced in our PIV data analysis). After calculation of MC for all test points of different FOVs and snapshots, model accuracies in terms of mass conservation is evaluated using conventional statistical metrics, namely MedAE and $R^2$-score.

Although the scores are calculable for each individual snapshot, they are calculated and averaged for various snapshots of each of the four FOVs in order to stabilize their trends and extract the general effect of various parameters in performance of each ML model. Our examinations of the present data indicated that after using 100 snapshots, the time-averaged metrics do not experience meaningful change and converge. So, in order to avoid unnecessary computational time, we sufficed 100 snapshots out of the entire set of 2000 snapshots for this averaging operation.

4. Results and discussion

Data pre-process (including extraction of flow-based features from the original PIV data, feature scaling and artificial gap generation) and post-process of the ML model outcome (including calculation of statistical and flow-based metrics) were carried out by scripting in python. The ML training and generalization for both MLP and SVR were performed using the open-source external package Scikit-learn [54].

Prior to hyper-parameter optimization (section 4.6), the following configuration of hyper-parameters is considered for two implemented ML models: $C = 50$, $\gamma = 1$ and $\varepsilon = 0.01$ for the SVR, and 32 and 64 neurons in the first and second hidden layers of the MLP architecture. There are also one output layer including one neuron (the target velocity component), and one input layer the number of neurons of which depends on the combination of applied features. For MLP training, based on a random initialization of the neural weights and a constant learning rate ($10^{-3}$), the weights are optimized using the stochastic gradient-based scheme (the ‘Adam’ optimizer in scikit-learn). The early stopping attribute in the validation procedure is applied to terminate training process when the validation score does not improve by a predefined tolerance ($10^{-4}$) for ten successive epochs.

4.1. Contribution of flow-based input features

The velocity components obtained by the ML models are evaluated for three different combinations of input features:

(a) Combination A: the $x$- and $y$-coordinate of each velocity component, i.e. ‘type A’ features (based on definitions of section 2.3)

(b) Combination B: the velocity vectors coordinate plus the filtered velocity components, i.e. ‘type A + type B’ features

(c) Combination C: the velocity components coordinate plus the filtered velocity plus the filtered vorticity/strain components, i.e. ‘type A + type B + type C’ features

Table 2 examines the effect of these three feature combinations on the accuracy of velocity component reconstructions by SVR and MLP models in term of MedNAE and $R^2$-scores. Our main concern is to investigate whether employing flow-based features in addition to the coordinate-based features enhances the prediction accuracy. Based on table 2, different trends are observed for SVR and MLP outcomes, particularly when combinations A and B are compared. More specifically, although adding the filtered velocity components as predictor features is quite beneficial in growth of MLP model accuracy, such configuration does not enhance the accuracy of SVR model. Low accuracy of SVR could be addressed by strong sensitivity of this ML model to the features selected and their scales [55]. It will be shown that utilization of appropriate scaling scheme circumvents such problem. Regarding the MLP model, negative values of $R^2$-score in combination A indicates that the coordinates as the input feature are not enough for this model to correctly predict the velocity field. However, adding the filtered velocity components as predictive features increases the $R^2$-score to above 0.75 for all velocity components and decreases the MedNAE by an order of magnitude. This indicates the important role of flow-based quantities as the predictors.

Table 2 also shows the sharp reduction of both model accuracies after inclusion of velocity deformation based features (the filtered vorticity/strain components). The orders of magnitude of vorticity and strain components are considerably different from those of velocity components and coordinate index. According to this issue and the results of combination C, it is necessary to scale the input features in order to make them more consistent and effective in achieving a better performance.

Table 3 evaluates the contribution of flow-based features to the accuracy of SVR and MLP reconstructions after applying various scalers on the data, namely the normalizer, min–max, standard and robust scalers. The results are compared for three combinations of input features: A, B and C. A comparison is also made with the predictions achieved without scaling procedure. A general enhancement of error metrics is evident as the result of feature scaling prior to the training procedure, particularly in combination C which includes features with a diverse range of values and patterns.

Comparison of results obtained for three different feature combinations in presence of scaling in terms of both $R^2$-score and MedNAE, indicates that combinations including flow-based features (combinations B and C) result in higher accuracies compared to the coordinate-based features (combination A). Among these flow-based configurations, the combination B which does not include velocity deformation based features (vorticity and strain components) considerably outperforms the combination C. A possible reason would be the lower fluctuations of ‘type B’ features (which are calculated after filtering the velocity field) in comparison with ‘type C’ features which are derivative-based quantities.

Based on the results presented in table 3, the standard, robust and min–max scalers cause significantly better scores compared with those of the normalizer scaler. Considering combination C as the active feature configuration, the
min–max scaler leads to more accurate reconstruction of velocity field. This is due to very different scales of vorticity/strain compared to types A and B features. The min–max scaler has a complete control to the output scaling range and consequently is more successful for those features in comparison with the standard and robust scalers that do scaling based on the distribution attributes of each feature. However, comparison of the best scores in table 3 (with bold figures) confirms that the combination B of features and utilization of standard scaler (and robust scaler by a little difference) lead to the most accurate reconstruction. High performance of standard scaler method is expected to be due to normal distribution of velocity data, while capability of robust scaler in diminishing the contribution of outlier data could be the main reason for successful prediction by this scaling method. In the following sections of the article, feature combination B processed by standard scaler are utilized.

4.2. Filtering

Two box filters, one with uniform and the other with Gaussian weights, are implemented in configuration B features and their effect as well as the contribution of filter length scale is presented in table 4 in terms of $R^2$-score. Each integer in the curly brackets indicates the filter size which is an integer multiplier to the PIV grid spacing in both $x$ and $y$ directions. For instance, $\{3, 5\}$ specifies two square filter boxes by dimensions $3\delta \times 3\delta$ and $5\delta \times 5\delta$. There are random artificial gappy squares underneath each filter box and throughout each FOV, as described in section 2.4, which are applied with a range of dimensions from $2 \times 2$ to $4 \times 4$.

The $R^2$-score in the case of $\{3, 5\}$ is significantly lower than the predictions made by the other filter scales. The minimum filter scale in the case of $\{3, 5\}$ is $3 \times 3$ which is smaller than the size of some gappy boxes and covers just the null values in such positions. In order to select a primary configuration from table 4 for the following sections, the absolute difference between each score with the maximum score in the same row is presented in table 5. Zero values indicate the maximum score in the corresponding row. The items with the difference smaller than 0.005 (0.5%) are specified in bold. Although most items with maximum score are in the $\{7\}$ configuration, the overall accuracy of $\{7, 9\}$ seems to be slightly better (particularly regarding the third velocity component predicted based on the Gaussian kernel) which is consequently utilized in the subsequent sections. The lower overall accuracy in the case of $\{5, 7, 9\}$ (excluding the $\{3, 5\}$ configuration) indicates that utilizing more filter-based features does not necessarily enhance the model performance.

In terms of filter kernel shape, table 4 shows that slightly better predictions are made by the Gaussian weights as compared with those of the uniform box filter. This is because of weight coefficient reduction in the Gaussian kernel away from the target point, while the uniform kernel considers similar weights for all in-box points.

### 4.3. Velocity field reconstruction

Figures 4–6 depict contour plots of the three instantaneous velocity components in the rotor exit region. Each of these three figures contains four parts: (a) the originally measured velocity field by SPIV; (b) a velocity map after artificial generation of numerous random gap boxes with dimensions ranging from $2 \times 2$ to $4 \times 4$ and 25% gappiness; (c) velocity field prediction by SVR; (d) the reconstructed velocity field by MLP.

All four FOVs are included in figures 4–6 because of their different flow attributes. The rotor exit-flow, which is dominant in FOVs 1 and 2, is significantly cyclic due to jet-wake flow structures after the blades. Moving downstream towards FOVs 3 and 4, the high-momentum jets and low-momentum wakes have mixed and produced more uniform flow patterns in the volute. These patterns are particularly evident from the in-plane velocity components (figures 4 and 5), while the out-of-plane velocity component has more complicated and fluctuating pattern due to three-dimensional effects in the rotor-exit flow of this type of turbomachine [41]. The diversity of characteristics mentioned for different velocity components at four FOVs provides an appropriate collection of data for ML model evaluations.

The MedNAE and $R^2$-scores for the SVR and MLP predictions are also presented in parts (c) and (d) of figures 4–6. In terms of these metrics, the SVR and MLP models have similar accuracies, slightly in favour of SVR. Comparison of identical iso-velocity contours in parts (c) and (d) of each figure with those of part (a) as the target indicates that MLP regressor generates more smooth velocity patterns, in
Table 3. Contribution of different scaling methods on the performance of SVR and MLP models in prediction of separate velocity components using various predictive feature configurations.

| Active features | ML model | Vel. comp. | Scaling method | None MedNAE $R^2$-score | Normalizer MedNAE $R^2$-score | Min–max scalar MedNAE $R^2$-score | Standard scalar MedNAE $R^2$-score | Robust scalar MedNAE $R^2$-score |
|----------------|----------|------------|----------------|---------------------------|-----------------------------|----------------------------------|-----------------------------------|----------------------------------|
| Combination A  | SVR      | 1          | None           | 0.053 0.413               | 0.047 0.454                 | 0.039 0.643                      | 0.022 0.846                       | 0.032 0.695                      |
|                |          | 2          |                | 0.180 0.410               | 0.163 0.380                 | 0.144 0.666                      | 0.075 0.878                       | 0.118 0.762                      |
|                |          | 3          |                | 0.512 0.399               | 0.658 0.150                 | 0.576 0.311                      | 0.458 0.572                       | 0.536 0.416                      |
|                | MLP      | 1          |                | 0.122 –2.790              | 0.050 0.465                 | 0.043 0.609                      | 0.028 0.816                       | 0.031 0.735                      |
|                |          | 2          |                | 0.251 –0.131              | 0.158 0.410                 | 0.127 0.704                      | 0.072 0.899                       | 0.074 0.876                      |
|                |          | 3          |                | 0.762 –0.018              | 0.665 0.162                 | 0.570 0.376                      | 0.452 0.619                       | 0.495 0.562                      |
| Combination B  | SVR      | 1          |                | 0.057 0.370               | 0.035 0.701                 | 0.011 0.851                      | 0.010 0.946                       | 0.011 0.942                      |
|                |          | 2          |                | 0.182 0.377               | 0.094 0.805                 | 0.035 0.945                      | 0.031 0.971                       | 0.034 0.965                      |
|                |          | 3          |                | 0.532 0.329               | 0.321 0.749                 | 0.227 0.834                      | 0.227 0.847                       | 0.228 0.817                      |
|                | MLP      | 1          |                | 0.015 0.910               | 0.035 0.737                 | 0.011 0.938                      | 0.011 0.950                       | 0.011 0.924                      |
|                |          | 2          |                | 0.066 0.908               | 0.094 0.826                 | 0.034 0.971                      | 0.034 0.975                       | 0.034 0.974                      |
|                |          | 3          |                | 0.366 0.755               | 0.322 0.743                 | 0.234 0.862                      | 0.244 0.841                       | 0.239 0.835                      |
| Combination C  | SVR      | 1          |                | 0.077 –0.019              | 0.049 0.468                 | 0.010 0.902                      | 0.018 0.725                       | 0.015 0.762                      |
|                |          | 2          |                | 0.245 –0.010              | 0.150 0.551                 | 0.032 0.957                      | 0.060 0.711                       | 0.050 0.733                      |
|                |          | 3          |                | 0.669 –0.008              | 0.480 0.466                 | 0.225 0.847                      | 0.340 0.591                       | 0.310 0.654                      |
|                | MLP      | 1          |                | 0.030 –2.180              | 0.044 0.568                 | 0.013 0.931                      | 0.011 0.928                       | 0.011 0.924                      |
|                |          | 2          |                | 0.114 0.417               | 0.120 0.719                 | 0.043 0.957                      | 0.035 0.970                       | 0.036 0.959                      |
|                |          | 3          |                | 0.645 –0.048              | 0.424 0.583                 | 0.245 0.860                      | 0.256 0.830                       | 0.255 0.834                      |
Table 4. Effect of different filter kernels and scales on the $R^2$-score of SVR and MLP models in prediction of separate velocity components.

| Filter kernel | ML model | Vel. comp. | $\{3, 5\}$ | $\{5, 7\}$ | $\{7, 9\}$ | $\{9, 11\}$ | $\{7\}$ | $\{5, 7, 9\}$ |
|---------------|----------|------------|-------------|-------------|-------------|-------------|--------|---------------|
| Uniform       | SVR      | 1          | 0.7415      | 0.9460      | 0.9554      | 0.9516      | 0.9595 | 0.9451        |
|               |          | 2          | 0.8023      | 0.9693      | 0.9783      | 0.9773      | 0.9799 | 0.9719        |
|               | MLP      | 1          | 0.5666      | 0.8431      | 0.8473      | 0.7874      | 0.8307 | 0.8198        |
|               |          | 2          | 0.5881      | 0.9718      | 0.9717      | 0.9661      | 0.9725 | 0.9684        |
|               |          | 3          | 0.5063      | 0.8390      | 0.8126      | 0.7713      | 0.8195 | 0.8414        |
| Gaussian      | SVR      | 1          | 0.7456      | 0.9458      | 0.9571      | 0.9584      | 0.9579 | 0.9465        |
|               |          | 2          | 0.8050      | 0.9710      | 0.9796      | 0.9805      | 0.9819 | 0.9724        |
|               | MLP      | 1          | 0.5063      | 0.8390      | 0.8126      | 0.7713      | 0.8195 | 0.8414        |
|               |          | 2          | 0.5764      | 0.9751      | 0.9747      | 0.9745      | 0.9729 | 0.9729        |
|               |          | 3          | 0.5506      | 0.8406      | 0.8398      | 0.7969      | 0.8096 | 0.8270        |

Table 5. The absolute difference between each score in table 4 with the maximum score existed in the same row.

| Filter kernel | ML model | Vel. comp. | $\{3, 5\}$ | $\{5, 7\}$ | $\{7, 9\}$ | $\{9, 11\}$ | $\{7\}$ | $\{5, 7, 9\}$ |
|---------------|----------|------------|-------------|-------------|-------------|-------------|--------|---------------|
| Uniform       | SVR      | 1          | 0.2181      | 0.0135      | 0.0041      | 0.0079      | 0      | 0.0145        |
|               |          | 2          | 0.1777      | 0.0106      | 0.0017      | 0.0026      | 0      | 0.0081        |
|               | MLP      | 1          | 0.2807      | 0.0042      | 0           | 0.0600      | 0.0167 | 0.0276        |
|               |          | 2          | 0.3844      | 0.0007      | 0.0008      | 0.0064      | 0      | 0.0041        |
|               |          | 3          | 0.3351      | 0.0024      | 0.0287      | 0.0701      | 0.0218 | 0             |
| Gaussian      | SVR      | 1          | 0.2128      | 0.0126      | 0.0013      | 0           | 0.0005 | 0.0119        |
|               |          | 2          | 0.1770      | 0.0109      | 0.0024      | 0.0015      | 0      | 0.0095        |
|               | MLP      | 1          | 0.2420      | 0.0039      | 0           | 0.0145      | 0.0174 | 0.0128        |
|               |          | 2          | 2.2066      | 0.0012      | 0.0006      | 0.0058      | 0      | 0.0029        |
|               |          | 3          | 0.3987      | 0           | 0.0004      | 0.0052      | 0.0004 | 0.0022        |
|               |          | 3          | 0.2899      | 0           | 0.0008      | 0.0437      | 0.0309 | 0.0136        |

comparison to the SVM model which reconstructs the original velocity patterns with more detail. From a ML viewpoint, precise reconstruction of the flow field patterns, particularly in cases with more fluctuative trends, could cause two drawbacks: (a) it makes the ML model more complicated and consequently rises the computational cost; (b) overfitting is more likely when a more complex model (namely SVM with more complicated kernel or MLP with deeper hidden layers and more number of neurons in each hidden layer) are utilized. However, regarding the fluid dynamics viewpoint, a more accurate reconstruction of the original wavy pattern is crucial, due to the sensitivity of velocity-derived quantities to reconstruction accuracy. Consequently, if the overfitting problem is avoided, a more complicated and more accurate ML model is plausible. This issue is confirmed in section 4.5 by comparing the scores calculated in the training, cross-validation and testing stages and also in section 4.6.2 by evaluating the accuracy in terms of mass conservation factor.

Possible size of reproducible gappy regions depends on the selected features and their availability in the target gappy locations. For combination B, with $7 \times 7$ and $9 \times 9$ filter dimensions, models are capable of reconstructing gappy regions as large as a $7 \times 7$ square box. According to figures 4–6, even at challenging locations of FOVs 1 and 4 with large spots of gappy data, ML models are successful in full reconstruction of the velocity field. More specifically, it is noticeable from the prediction for the bottom right corner of FOV 1 in figure 4 and the top left corner of FOV 4 in figure 5 that the ML models have successfully reconstructed the jet-wake flow pattern in a consistent way, although most of data in those regions were originally missing.

Figure 7 illustrates reconstruction of three velocity components of an instantaneous snapshot of FOV 2 with sparse data and gapiness of 50% including random gap boxes with dimensions from $2 \times 2$ to $6 \times 6$. Considering the extensive regions with artificial null data, the overall capability of SVR and MLP models in predicting a similar pattern to that of the
original velocity is still satisfactory, and slightly in favour of the SVR model. There are some local jumps in the predicted values, particularly in the MLP model. They are the dark or bright spots marked in part (d) of figure 7 for the x-component velocity of MLP reconstruction. Based on their corresponding location marked in part (b) of figure 7, all the four spots are completely surrounded by clustered null data. This deteriorates the reconstruction quality of the ML models. However, assessment of SVR results (part (c) of the figure) indicates that this model is more tolerable to such clustered missing data and provides smoother reconstruction in such regions. It should be mentioned that after applying a larger filter scale in the filter-based feature extraction, these discontinuities disappear due to improved data density in the filter box, though this could reduce the overall accuracy of the regressor.

In order to examine the velocity profile predicted by the ML models in more quantitative manner, figure 8 demonstrates the output of figure 7 evaluated at the horizontal dashed line shown in that figure (the line \( y = 40 \delta \), where \( \delta \) is the PIV grid spacing and the coordinate origin is located at the lower left corner of the FOV). This figure compares SVR and MLP predictions of velocity (solid curves) with the original velocity as well as the artificially gap-augmented data (scattered symbols). The velocity at locations with originally missing data (due to PIV data validation) or locations selected as artificial gaps (for model evaluation) is zero and otherwise, it is the same as the original measured velocity. The reproduction of the true in-plane velocity profile in the gappy points is considerably more accurate than the reconstruction of the out-of-plane velocities. This is in agreement with the scores and trends presented previously in figures 4–6 for lower level of gappiness percentage. The wavier pattern of the out-of-plane velocity component is the main reason for achieving a lower accuracy of reconstruction.

Despite alternating pattern of the y-component of velocity and large bundles of artificial missing data, namely between \( x = 23 \delta \) and \( x = 35 \delta \), both SVM and MLP have been quite successful in prediction of the velocity peak at around \( x = 30 \delta \) and reconstruction of the steep changes at its side regions. Furthermore, the discontinuity in the x-component velocity profile induced by gappiness of the original data between \( x = 55 \delta \) and \( x = 57 \delta \) is located at a high-gradient region of the velocity field. The predicted trend by the ML models at this interval is accurate. Both predictions show local peaks which indicate large scale turbulence which are not evident in point to point velocity values but were also repeated in iso-velocity contours.

**4.4. Assessment of the gap effect**

Capability of SVR and MLP in reconstruction of two-dimensional instantaneous vector maps in FOV2 is investigated in figure 9 for two extensively sparse fields with
different levels of gappiness percentage, namely 50% and 70%. Such broadly sparse missing data are possible for PIV data of complex flow fields, when seeding concentration is non-homogeneous and insufficient in some clustered regions. The blue vectors in figure 9 indicate the original measured velocity, while the red ones denote the reconstructed vectors by the ML models. Although growth of gappiness slightly increases the deviation between true and predicted vectors, reasonable coincidence of vectors, even for largest amount of gappiness, indicates good potential of both ML models in correct prediction of vector maps. However, more detailed comparison of the results indicates marginally better coincidence for the vectors obtained by SVR than those of MLP. This is also confirmed by the $R^2$-scores reported in figure 9 for each vector plot calculated based on the planar velocity magnitude.

Table 6 presents the effect of gap percentage on the accuracy of SVR and MLP regressors in terms of $R^2$-score for different velocity components. Generally the score for both SVR and MLP approaches decrease with increase of the gap percentage, which is expected. An accuracy reduction rate ($ARR_{\alpha-\beta}$) is defined as the relative decrease of $R^2$-score by growing the gap percentage from $\alpha\%$ to $\beta\%$, given as:

$$ARR_{\alpha-\beta} = \frac{RS_\alpha - RS_\beta}{RS_\alpha} \times 100\%$$

where $RS_\alpha$ and $RS_\beta$ are $R^2$-scores for $\alpha\%$ to $\beta\%$ gaps, respectively. Positive values of accuracy reduction rate correspond to relative reduction of $R^2$-score by increase of the gap percentage. Comparison of $ARR_{15-60}$ and $ARR_{60-75}$ shows that accuracy reduction is minimum for a 45% growth in gappiness from 15% to 60% while it is significantly higher (particularly for MLP reconstruction) when the gap is further enlarged by just 15%.

4.5. Hyper-parameter optimization

The hyper-parameters utilized up to here, were the same for all presented results (as introduced in section 4.1). However, in order to achieve the best predictions by SVR and MLP regressors, their main hyper-parameters are optimized with randomized cross-validation search. The optimal condition is searched such that it maximizes the $k$-fold cross-validation score. The optimized parameters include the penalty coefficient $C$, the kernel coefficient $\gamma$ and the $\epsilon$ parameter in the SVR model and the number of neurons ($N_1, N_2, N_3$) for MLP with one to three hidden layers.

Tables 7 and 8 present the best hyper-parameters for SVR and MLP models averaged for the four FOVs and 100 snapshots. These tables also give the train, test and mean and standard deviation of $k$-fold cross-validation scores. For MLP, three architectures are considered in table 8, such that the single-, double- or triple-layer perceptron corresponds to a
MLP with one, two or three hidden layers, respectively. Performance of various kernels of the SVR and different activation functions of the MLP regressors were also evaluated in the hyper-parameter optimization. The linear, polynomial, RBF and sigmoid kernels were considered for SVR and the identity, hyperbolic tangent, ReLU and logistic activation functions were assessed for the MLP to find the optimal functions. In almost all configurations presented in tables 7 and 8, the RBF and ReLU were the best kernel and activation functions, respectively. Based on comparison of these tables, the SVR scores are generally better than values obtained by MLP for all velocity components. Furthermore, both the SVR and MLP models lead to significantly better accuracies in prediction of the in-plane velocity components (with test $R^2$-scores higher than 0.989 for SVR and 0.97 for two-layer perceptron) in comparison with that of out-of-plane component (with test $R^2$-score that is at most 0.922 for SVR and 0.898 for two-layer perceptron).

In order to check the state of overfitting in the training process, the difference between $R^2$-score of train and test procedures are presented in the last column of tables 7 and 8. Smallness of train-test score differences confirms the overall absence of overfitting. However, considerably higher values of train-test score difference for the out-of-plane velocity component with respect to the in-plane components indicate that the former has higher potential for overfitting. This could be because of a wavier field for the out-of-plane component. Moreover, comparison of train-test difference for the MLP model for each velocity component shows that using larger number of layers increases that difference and adds to the higher risk of overfitting.

4.6. Statistical and physical analysis of reconstruction errors

4.6.1. Statistical evaluation based on the velocity components. In order to analyze the reconstruction error by SVR and MLP for all locations of an instantaneous snapshot, figure 10 illustrates the boxplot of normalized absolute error ($\Delta U_i^*$, the left frames) as well as absolute error boxplots ($\Delta U_i$, the right frames) calculated based on the following equations:

$$\Delta U_i = |U_{\text{ML}}^i - U_{\text{PIV}}^i|, \quad i = 1, 2, 3$$  \hspace{1cm} (23)

$$\Delta U_{\text{PIV}}^i = \frac{\Delta U_i}{U_{\text{PIV}}^i}, \quad i = 1, 2, 3$$  \hspace{1cm} (24)

where $U_{\text{PIV}}^i$ and $U_{\text{ML}}^i$ are each of three instantaneous velocity components, measured by PIV or reconstructed by ML regressors, respectively. In each frame, SVM and MLP predictions are compared for different FOVs. The deviation in prediction of the out-of-plane velocity component is one order of magnitude higher than those of the in-plane components in terms of the normalized absolute error. Two reasons are responsible for such trend. First, there is wavier and more...
complicated field with no general pattern for the third velocity component compared to that of the in-plane components, so it does not follow its neighbouring velocities or any general pattern that might be traced in the snapshot. This causes an extended absolute error between prediction output and target values for the out-of-plane velocity component in comparison with those of planar components (compare the interquartile ranges for the frames in the right column of figure 10 for different velocity components). The second reason is considerably lower values of the third velocity component compared to the in-plane components (see figures 4–6), that enhances the normalized absolute error due to its definition (division by the velocity component in equation (24)). Examination of AE and NAE values as the absolute and relative metrics of the model accuracy shows that in most cases (among different FOVs and velocity components) SVR prediction has resulted in marginally smaller errors. This is the trend in terms of either median, Q1 or Q3 of the boxplots. The normalized absolute errors obtained for FOV1 for the planar velocity components (x and y components), are larger but more scattered in contrast to comparable relative error bands in the second to forth FOVs. The dominance of normalized errors for the y-component velocity field in FOV1 is due to the existence of near-zero values for this velocity component (see part (a) of figure 5).

4.6.2. Physical evaluation based on the mass conservation factor. All the previous sections evaluated the ML model accuracy in terms of statistics calculated based on the velocity components. Consistency of the reconstructed flow field with the original PIV field regarding mass flux conservation is also quite important from fluid mechanics viewpoint. This is evaluated by statistics calculated based on mass conservation factor that we proposed in section 3. Figure 11 illustrates scatter plot of predicted vs. target mass conservation factor for different FOVs. The solid line with a slope of 45° in each plot indicates the exact modeling. More proximity of scattered points to this line indicates more similar ML-based reconstruction of mass flux characteristics in comparison with those obtained from the original PIV flow map as the target. It is obvious that the SVR prediction outperforms the MLP model in this regard.

In order to evaluate the accuracy of MC reproduction by ML models in comparison with that of the original PIV field, different error statistics including MAE, root mean squared error (RMSE), MedAE and $R^2$-score are computed which are presented at the top of each frame in figure 11. For each
statistics, two values are given in the parenthesis corresponding to SVR and MLP models, respectively. Comparison of these pair of values indicates considerably better performance of SVR than MLP in terms of all error metrics. It should be also noticed that $R^2$-score calculated here based on mass conservation factor is less than those previously evaluated based on in-plane velocity components (tables 7 and 8). The reason for this reduction of accuracy is derivative-based definition of MC that significantly intensifies inaccuracy of in-plane velocity component predictions. This decrease is more pronounced in the case of MLP model due to its lower accuracy in reconstruction of velocity components.

Figure 8. Comparison of SVR and MLP predictions of velocity (solid curves) with the original and artificially gapped velocity (scattered symbols) for (a) $x$-component, (b) $y$-component and (c) $z$-component with respective subscripts 1–3.
Figure 9. SVR and MLP reconstruction of two-dimensional instantaneous vector maps compared with the true velocity vectors in FOV2 for two different levels of gappiness percentage: (a) 50%; (b) 70%.

Table 6. Effect of gap percentage on the $R^2$-score of SVR and MLP models in prediction of separate velocity components.

| ML model | Vel. comp. | Gap percentage | Accuracy reduction rate |
|----------|------------|----------------|------------------------|
|          |            | 15% | 30% | 45% | 60% | 75% | AR$R_{15-60}$ | AR$R_{60-75}$ |
| SVR      | 1          | 0.963 | 0.957 | 0.948 | 0.931 | 0.889 | 3.34 | 4.46 |
|          | 2          | 0.979 | 0.977 | 0.974 | 0.964 | 0.937 | 1.52 | 2.81 |
|          | 3          | 0.852 | 0.855 | 0.819 | 0.755 | 0.660 | 11.36 | 12.54 |
| MLP      | 1          | 0.950 | 0.950 | 0.939 | 0.899 | 0.787 | 5.31 | 12.55 |
|          | 2          | 0.970 | 0.971 | 0.966 | 0.955 | 0.877 | 1.57 | 8.11 |
|          | 3          | 0.839 | 0.844 | 0.807 | 0.757 | 0.639 | 9.67 | 15.61 |

Although based on the $R^2$-scores presented in figure 11, close accuracies are calculated for different FOVs, other metrics indicate higher absolute errors for the near-rotor regions, namely FOVs 1 and 2. Our previous studies indicated that the flow field at these FOVs involves higher levels of three-dimensionality [41, 43]. This increases the out-of-plane mass flux and consequently elevates the mass conservation factor (as is evident from figure 11 based on the range of MC) and its corresponding absolute error. Figure 12 illustrates box plot of the relative reconstruction error of mass conservation.
Table 7. Optimal hyper-parameters for SVR model including training, testing and cross-validation scores.

| Vel. comp. | C    | γ   | ε    | Train \( R^2 \)-score | Mean of CV \( R^2 \)-score | Std. dev. of CV \( R^2 \)-score | Test \( R^2 \)-score | Train-test \( R^2 \)-score difference |
|------------|------|-----|------|------------------------|----------------------------|-------------------------------|-------------------|-----------------------------------|
| 1          | 48.788 | 16.894 | 0.003 | 0.9996                | 0.9945                     | 0.0041                      | 0.9896            | 0.010                             |
| 2          | 48.153 | 8.725 | 0.014 | 0.9996                | 0.9961                     | 0.0020                      | 0.9934            | 0.006                             |
| 3          | 28.996 | 4.508 | 0.038 | 0.9893                | 0.9702                     | 0.0035                      | 0.9221            | 0.067                             |

Table 8. Optimal hyper-parameters for MLP model including training, testing and cross-validation scores.

| MLP architecture | Vel. comp. | \( N_1 \) | \( N_2 \) | \( N_3 \) | Train \( R^2 \)-score | Mean of CV \( R^2 \)-score | Std. dev. of CV \( R^2 \)-score | Test \( R^2 \)-score | Train-test \( R^2 \)-score difference |
|------------------|------------|----------|----------|----------|------------------------|----------------------------|-------------------------------|-------------------|-----------------------------------|
| Single-layer perceptron | 1 | 20 | NA | NA | 0.9398 | 0.9395 | 0.0051 | 0.9361 | 0.004 |
|                  | 2 | 52 | NA | NA | 0.9453 | 0.9443 | 0.0030 | 0.9424 | 0.003 |
|                  | 3 | 132 | NA | NA | 0.8381 | 0.8322 | 0.0278 | 0.8167 | 0.021 |
| Double-layer perceptron | 1 | 47 | 71 | NA | 0.9758 | 0.9751 | 0.0058 | 0.9716 | 0.004 |
|                  | 2 | 18 | 71 | NA | 0.9850 | 0.9840 | 0.0072 | 0.9809 | 0.004 |
|                  | 3 | 65 | 75 | NA | 0.9373 | 0.9158 | 0.0142 | 0.8983 | 0.039 |
| Triple-layer perceptron | 1 | 37 | 61 | 67 | 0.9836 | 0.9799 | 0.0090 | 0.9688 | 0.015 |
|                  | 2 | 24 | 64 | 50 | 0.9830 | 0.9819 | 0.0147 | 0.9757 | 0.007 |
|                  | 3 | 48 | 47 | 62 | 0.9422 | 0.9310 | 0.0223 | 0.8772 | 0.065 |

factor (\( \Delta MC^* \)) for different FOVs and the two ML models. \( \Delta MC^* \) is defined as follows:

\[
\Delta MC^* = \frac{|MC_{PIV} - MC_{ML}|}{\sigma_{MC,PIV}}
\]  

(25)

where \( \sigma_{MC,PIV} \) is the standard deviation of MC calculated for the test data of each particular FOV. Figure 12 indicates that after normalization of the absolute error of MC, the errors for each of the ML models are comparable in different errors of FOV. In contrast, the median of relative error calculated for MLP is approximately 0.3\( \sigma_{MC,PIV} \), which is up to three times larger than that of the SVR model. This is also valid for the interquartile range that expresses considerably higher variance of MC-based error for the MLP model in comparison with that of the SVR.

4.7. ML models performance vs. conventional spatial interpolators

It is necessary to evaluate the reconstruction performance of the present ML models in comparison with the other conventional statistical approaches. Six additional methods are employed in order to reconstruct the missing data of the PIV field, as briefly introduced in the following:

(a) GPOD: As a generation of POD method, this is widely used to reconstruct the missing data of time-resolved PIV [16, 17, 56]. Based on modal decomposition of velocity field according to their contribution to the total kinetic energy, GPOD is capable to either reduce the dimensionality of the complex non-linear flow phenomena or reconstruct the velocity field. Sufficient number of temporally-correlated snapshots should be utilized to extract meaningful data for the gappy regions. The details of GPOD algorithm is presented in appendix.

(b) Nearest-neighbor interpolation (NNI): In this simple spatial interpolation algorithm, the approximated value of the target quantity at any arbitrary location is considered to be identical to the sample data which is closest to that location [57].

(c) Triangulation with linear interpolation (TLI): This method is based on a triangular network created from the sampled grid nodes. Each un-sampled node locates in a triangle with known values at its three vertices [58].

(d) Uniform-weighted moving average interpolation (UMAI): This is a simple averaging of the sampled data located in the averaging region. For each un-sampled data, a square region centred by the coordinate of that node is utilized for interpolation [59].

(e) Gaussian-weighted moving average interpolation (GMAI): In comparison with the UMAI that applies uniform weights to all sampled data inside the averaging region, the GMAI employs a Gaussian filter to generate different weights for various sampled data based on their location with respect to the central un-sampled data.
Figure 10. Box plots of prediction error for different velocity components: (a) the $x$-component; (b) the $y$-component; (c) the $z$-component. The left and right frames present the normalized absolute error and the absolute error of the prediction output with respect to the target (PIV measurements), respectively.

In fact, this method could consider the stronger contribution of the nearer sampled data in the averaging process by a larger weight [59].

(f) Moving median interpolation (MMI): Median filter provides a strong noise reduction capability and is robust to outliers in either time series or spatial fields [60]. By a similar procedure to the UMAI, here the median of sampled data in a square region centred by the un-sampled node is calculated as the interpolated value.

Figure 13 illustrates the scatter plot of predicted vs. target velocity component in $x$ direction for two ML models (SVR and MLP) and six interpolators. For this figure and the subsequent ones in this section, the data belongs to FOV2 with
Figure 11. Scatter plot of predicted vs. target mass conservation factor for the four FOVs. The FOV number and MC reconstruction accuracy in terms of MAE, RMSE, MedAE and $R^2$-score are given at the top of each plot.

Figure 12. Box plots of normalized absolute error of the mass conservation factor.
Figure 13. Scatter plot of predicted vs. target velocity component in x direction, based on two ML models and six interpolators. The predictor name and velocity reconstruction accuracy in terms of MANE, RMSNAE, MedNAE and $R^2$-score are given at the top of each subplot.
Figure 14. Box plots of normalized absolute error of prediction performed by eight predictors for: (a) the $x$-component velocity; (b) the $y$-component velocity; (c) the $z$-component velocity; (d) the mass conservation factor.

50% gap percentage. The solid line with the 45° slope in each subplot indicates the exact modeling. Different error statistics including mean, root mean square and median of the normalized absolute error (MNAE, RMSNAE and MedNAE, respectively), and $R^2$-score are computed which are presented at the top of each frame in figure 13. SVR leads to the best accuracy in terms of all statistics. The $R^2$-score obtained for the MLP and MMI are approximately the same, and slightly smaller than those computed for the moving-average based predictors (GMAI and UMAI). Comparable accuracy is also obtained by the TLI interpolator. In contrast, the results achieved for the NNI almost shows uncorrelated reconstruction with respect to the true velocities. This could be due to noticeably deviatoric value of the first available neighboring node with respect to the gappy node when there is a considerable distance between these two nodes (this especially happens for large clustered gaps).

Figure 13 also shows that the GPOD method leads to a small accuracy in terms of $R^2$-score compared to the other predictors (except the NNI approach). The GPOD method needs numerous spatio-temporal correlated snapshots to construct the appropriate velocity field at the gappy regions. Although the accuracy of this model is well for high-frequency (time-resolved) velocity data, it works weakly for insufficient temporal resolutions or large spatial gappiness [61]. More specifically, the correlation matrix calculated in the iterative procedure of GPOD (from equation (A5) in appendix) strictly depends on the temporal correlation between various snapshots. The low-frequency of the PIV system in the present study considerably reduces this correlation and decreases the reconstruction accuracy of GPOD method.

Figure 14 shows the box plots of normalized absolute error of prediction performed by the eight predictors for the three velocity components as well as the mass conservation factor. Considerably larger errors observed for the NNI and GPOD methods, either in terms of different velocity components or mass conservation metric, confirm that these two predictors are not suitable for instantaneous reconstruction of velocity field, at least for low-frequency data or the spatial data with extensive gaps. Based on the normalized absolute error, SVR is again the most accurate model, MLP and TLI have comparable error statistics that are larger than SVR, and the moving average/median methods experience the third error level.

Figure 15 evaluates the performance of two ML models in comparison with the six additional predictors based on scatter plots of predicted vs. target mass conservation factor. The SVR
clearly outperforms the other predictors, and MLP and TLI present the second accuracy level. The moving average based methods produce slightly better reconstruction in terms of MC index compared to that of the MMI. Also, a marginally higher accuracy can be observed for various error statistics of the GMAI in comparison with the UMAI, that could be due to non-uniform Gaussian weights of the former method characterizing the contribution of samples distance with respect to
the central gappy node. This presents a clear step forward in application of ML plus fluid mechanics validity index in velocity field reconstruction. The index gives an indication of how conventional reconstruction methods could deviate in producing physically meaningful velocity fields.

5. Conclusions

A ML approach was established for reconstruction of the measured velocity data of jet/wake flow in the rotor exit region of a centrifugal turbomachine. The SVR and MLP algorithms were chosen for such purpose due to their high potential in handling the non-linear flow phenomena, robustness against overfitting and flexibility in learning procedure. In addition to the conventional coordinate-based features, we proposed practical flow-based features with fundamental fluid mechanics characteristics as the input space in order to enhance the reconstruction accuracy. This opens the way to select correct features that can connect data-based and physics-based models. Train, validation and test data sets were randomly separated after adding numerous artificial gaps with pre-defined size and overall gappiness percentage. The SVR and MLP hyper-parameters were optimally tuned by randomized search, maximizing the $k$-fold cross-validation score. In addition to velocity-based scores, a metric based on mass conservation was proposed to evaluate the capability of ML models in reconstruction of a physically meaningful flow field regarding fluid dynamics characteristics. The results also compared with the reconstructed velocity field by six conventional spatial interpolation methods. The main outcomes of the present study are summarized as follows:

- Implementation of flow-based features that highly correlate with velocity field was quite effective in enhancement of ML-based reconstructions. Examination of different combinations of the generated features showed that a combination including filtered velocity components (based on a Gaussian filter) and the coordinate-based features leads to the highest accuracy in velocity reproduction. Utilization of derivative-based features (filtered vorticity/strain components) was not as effective as the filtered velocity-based features due to their fluctuative behaviour.
- Implementation of different scaling approaches namely normalizer, min–max scaler, standard scaler and robust scaler proved higher performance of the standard scaler due to normal distribution of velocity field data, and competitive function of the robust scaler due to isolating outlier data.
- The proposed approach was able to reproduce extensively clustered missing data by utilizing a larger filter scale. SVR and MLP models satisfactorily reconstructed velocity field for extremely sparse data. This was confirmed after reproduction of reasonable vector plots from these models as compared with the original vector maps by $R^2$-score as high as 0.94 for the in-plane velocity components and for gappiness levels up to 75%.
- The SVR reconstruction accuracy is slightly superior to that of the MLP outcome, in terms of either median normalized absolute error or $R^2$-score, for all velocity components and different levels of gappiness.
- After hyper-parameter optimization, the $R^2$ scores for SVR reached to 0.989 for in-plane and 0.922 for out of plane velocity components in comparison with respective scores of 0.97 and 0.898 for MLP.
- The proposed mass conservation metric produced a more distinct advantage for SVR. More specifically, the MC-based $R^2$-scores of SVR and MLP were up to 0.96 and 0.79, respectively and the median of MC-based relative error obtained for MLP was up to three times larger than that of the SVR model. Furthermore, the MC-based $R^2$-scores obtained for both models was smaller than the velocity-based $R^2$-scores, indicating more sensitive behaviour of MC-based statistics that is because of the derivative-based definition of mass conservation factor.
- Comparison of the two ML models outcome with the estimations performed by six additional spatial interpolation methods indicated the superior performance of SVR with respect to all the methods employed, either in terms of velocity based errors or MC-based scores. The accuracy of MLP was comparable to that of the TLI, and considerably higher than the moving average/median methods. The gappy POD and NNI approaches were not successful in accurate prediction of the velocity field, due to low-frequency of the PIV measurements or extensive clustered gappy regions in the flow domain.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Appendix

The GPOD method for reconstruction of the gappy data in velocity snapshots is inspired by the iterative repair procedure introduced by Everson and Sirovich [15]. A matrix $\mu$ is initially defined to address the location of gappy points:

$$\mu(n,m;k) = \begin{cases} 1 & \text{if } U_i(n,m;k) \text{ exists} \\ 0 & \text{if } U_i(n,m;k) \text{ is missing or erroneous} \end{cases} \quad (A1)$$

where $k$ is the snapshot number indicator and $n$ and $m$ notify spatial position of each node in the discretized network. The ensemble average of velocity ($\langle U_i \rangle : i = 1, 2, 3$) are employed as the initial guess to fill the gappy regions:

$$U_i(n,m;k) = \begin{cases} U_i(n,m;k) & \text{if } \mu(n,m;k) = 1 \\ \langle U_i \rangle (n,m) & \text{if } \mu(n,m;k) = 0 \end{cases} \quad (A2)$$
The GPOD algorithm is established on the basis of spatial eigenmodes, $\phi_i^{(s)}(n,m)$, calculated from snapshot POD method proposed by Sirovich [62]. The orthogonal eigenmodes are considered as linear combination of the velocity at different snapshots:

$$\phi_i^{(s)}(n,m) = \sum_k \alpha_i^{(s)}(k) U_i(n,m;k) \quad (A3)$$

where $\alpha$ is the snapshot eigenfunction, $s$ notifies the mode number and $i$ represents component of the velocity and eigenmode. The intermediate eigenvalue problem is expressed as:

$$\sum_k C_{k,l} \alpha_i^{(s)}(k) = \lambda_i^{(s)} \alpha_i^{(l)} \quad (A4)$$

where $\lambda$ is the eigenvalue and $l$ is the snapshot number indicator. The correlation matrix is constructed by the elements $C_{k,l}$ expressed as:

$$C_{k,l} = \frac{1}{K} \sum_n \sum_m U_i(n,m;k) U_l(n,m;l) \quad (A5)$$

where $K$ is the total number of used snapshots. Solving equation (A4), provides the snapshot eigenfunctions, $\alpha_i^{(s)}(k)$ and then using equation (A3), the spatial eigenmodes, $\phi_i^{(s)}(n,m)$, are calculable.

The iterative GPOD procedure is as follows:

(a) The snapshot POD is used to calculate the spatial eigenmodes, $\phi_i^{(s)}(n,m)$, using equations (A3)–(A5).

(b) An estimate of the velocity field, $\bar{U}_i(n,m;k)$, is constructed using an unknown set of orthonormal eigenfunctions, $b_i^{(s)}(k)$, with $q$ terms:

$$\bar{U}_i(n,m;k) = \sum_{s=1}^{q} b_i^{(s)}(k) \phi_i^{(s)}(n,m). \quad (A6)$$

(c) A mean square error, $E(k)$, is computed between the actual and estimated velocities at spatial locations where real data exists. The mask $\mu(n,m;k)$ ensures that only points with real data are included in the summation:

$$E(k) = \sum_n \sum_m E(n,m;k) \mu(n,m;k) \quad (A7)$$

where

$$E(n,m;k) = (\bar{U}_i(n,m;k) - U_i(n,m;k))^2. \quad (A8)$$

(d) A minimum mean square error is calculable after differentiating equation (A7) with respect to the unknown coefficients, $b_i^{(s)}(k)$, and equating the result to zero. This leads to a system of linear equations the solution of which provides the unknown coefficients, $b_i^{(s)}(k)$:

$$[M]_x^{l',s} [B]_{k,x} = [V]^{l',s} \quad (A9)$$

where

$$[M]^{l',s} = \sum_n \sum_m \phi_i^{(s)}(n,m) \phi_i^{(l')}^{(s)}(n,m) \mu(n,m;k)$$

$$[B]^{l',s}_k = b_i^{(s)}(k)$$

$$[V]^{l',s}_k = \sum_n \sum_m U_i(n,m;k) \phi_i^{(s)}(n,m) \mu(n,m;k). \quad (A10)$$

(e) Now, the calculated coefficients $b_i^{(s)}(k)$ are employed in equation (A6) to estimate the velocity field.

(f) The missing data is filled at the gappy points with the velocity estimates obtained in the previous step.

$$U_i(n,m;k) = \begin{cases} U_j(n,m;k) & \text{if } \mu(n,m;k) = 1 \\ U_l(n,m;k) & \text{if } \mu(n,m;k) = 0 \end{cases}. \quad (A11)$$

(g) Convergence of the velocity reconstruction is evaluated, and if it is not satisfied, the procedure iterates by returning to step one.

ORCID ID
Ghasem Akbari © https://orcid.org/0000-0001-8350-6901

References

[1] Raffel M, Willert C E, Scarano F, Kähler C J, Wereley S T and Kompenhans J 2018 Particle Image Velocimetry: A Practical Guide (Berlin: Springer)
[2] Vlasenko A, Steele E C and Nimmo-Smith W A M 2015 A physics-enabled flow restoration algorithm for sparse PIV and PTV measurements Meas. Sci. Technol. 26 065301
[3] Garcia D 2011 A fast all-in-one method for automated post-processing of PIV data Exp. Fluids 50 1247–59
[4] Tang C, Sun W, He H, Li H and Li E 2017 Spurious PIV vector detection and correction using a penalized least-squares method with adaptive order differentials Exp. Fluids 58 81
[5] Okuno T, Sugi Y and Nishio S 2000 Image measurement of flow field using physics-based dynamic model Meas. Sci. Technol. 11 667
[6] Suzuki T 2012 Reduced-order Kalman-filtered hybrid simulation combining particle tracking velocimetry and direct numerical simulation J. Fluid Mech. 709 249–88
[7] Westerweel J and Scarano F 2005 Universal outlier detection for PIV data Exp. Fluids 39 1096–100
[8] Duncan J, Dabiri D, Hove J and Gharib M 2010 Universal outlier detection for particle image velocimetry (PIV) and particle tracking velocimetry (PTV) data Meas. Sci. Technol. 21 057002
[9] Keane R D and Adrian R J 1990 Optimization of particle image velocimeters. I. Double pulsed systems Meas. Sci. Technol. 1 1202
[10] Lee Y, Yang H and Yin Z 2017 Outlier detection for particle image velocimetry data using a locally estimated noise variance Meas. Sci. Technol. 28 035301
[11] Nogueira J, Lecuona A and Rodriguez P 1997 Data validation, false vectors correction and derived magnitudes calculation on PIV data Meas. Sci. Technol. 8 1493
[12] Hart D P 2000 PIV error correction Exp. Fluids 29 13–22
[13] Pun C-S, Susanto A and Dabiri D 2007 Mode-ratio bootstrapping method for PIV outlier correction Meas. Sci. Technol. 18 3511
[14] Gunes H and Rist U 2007 Spatial resolution enhancement/smoothing of stereo–particle-image-velocimetry data using proper-orthogonal-decomposition–based and Kriging interpolation methods Phys. Fluids 19 064101

[15] Everson R and Sirovich L 1995 Karhunen–Loève procedure for gappy data J. Opt. Soc. Am. A 12 1657–64

[16] Venturi D and Karniadakis G E 2004 Gappy data and reconstruction procedures for flow past a cylinder J. Fluid Mech. 519 315–36

[17] Rabault J, Charonko J J and Vlachos P P 2012 Adaptive gappy proper orthogonal decomposition for particle image velocimetry data reconstruction Meas. Sci. Technol. 23 025303

[18] Wang H, Gao Q, Feng L, Wei R and Wang J 2015 Proper orthogonal decomposition based outlier correction for PIV data Exp. Fluids 56 43

[19] Higham J, Brevis W and Keylock C J 2016 A rapid non-iterative proper orthogonal decomposition based outlier detection and correction for PIV data Meas. Sci. Technol. 27 125203

[20] Saini P, Arndt C M and Steinberg A M 2016 Development and evaluation of gappy-POD as a data reconstruction technique for noisy PIV measurements in gas turbine combustors Exp. Fluids 57 122

[21] Wen X, Li Z, Peng D, Zhou W and Liu Y 2019 Missing data recovery using data fusion of incomplete complementary data sets: a particle image velocimetry application Phys. Fluids 31 025105

[22] Fukami K, Fukagata K and Taira K 2020 Assessment of supervised machine learning methods for fluid flows Theor. Comput. Fluid Dyn. 34 497–519

[23] Liang J, Cai S, Xu C and Chu J 2020 Filtering enhanced tomographic PIV reconstruction based on deep neural networks IET Cyber-Syst. Robot. 4 43–52

[24] Gao Q, Li Q, Pan S, Wang H, Wei R and Wang J 2019 Particle reconstruction of volumetric particle image velocimetry with strategy of machine learning (arXiv:1909.07815)

[25] Rabault J, Kolaas J and Jensen A 2017 Performing particle image velocimetry using artificial neural networks: a proof-of-concept Meas. Sci. Technol. 28 125201

[26] Cai S, Zhou S, Xu C and Gao Q 2019 Dense motion estimation of particle images via a convolutional neural network Exp. Fluids 60 73

[27] Cai S, Liang J, Gao Q, Xu C and Wei R 2019 Particle image velocimetry based on a deep learning motion estimator IEEE Trans. Instrum. Meas. 68 2538–54

[28] Gim Y, Jang D K, Sohn D K, Kim H and Ko H S 2020 Three-dimensional particle tracking velocimetry using shallow neural network for real-time analysis Exp. Fluids 61 1–8

[29] Barwey S, Hassanaly M, Raman V and Steinberg A 2019 Using machine learning to construct velocity fields from OH–PLIF images Combust. Sci. Technol. 1–24

[30] Tombul H, Ozbayoglu A M and Ozbayoglu M E 2019 Computational intelligence models for PIV based particle (cuttings) direction and velocity estimation in multi-phase flows J. Pet. Sci. Eng. 172 547–58

[31] Deng Z, Chen Y, Liu Y and Kim K C 2019 Time-resolved turbulent velocity field reconstruction using a long short-term memory (LSTM)-based artificial intelligence framework Phys. Fluids 31 075108

[32] Jin X, Laima S, Chen W-L and Li H 2020 Time-resolved reconstruction of flow field around a circular cylinder by recurrent neural networks based on non-time-resolved particle image velocimetry measurements Exp. Fluids 61 1–23

[33] Giannopoulos A and Aider J-L 2020 Prediction of the dynamics of a backward-facing step flow using focused time-delay neural networks and particle image velocimetry data-sets Int. J. Heat Fluid Flow 82 108533

[34] Liu B, Tang J, Huang H and Lu X-Y 2020 Deep learning methods for super-resolution reconstruction of turbulent flows Phys. Fluids 32 025105

[35] Deng Z, He C, Liu Y and Kim K C 2019 Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework Phys. Fluids 31 125111

[36] Labonté G 2001 Neural network reconstruction of fluid flows from tracer-particle displacements Exp. Fluids 30 399–409

[37] Pruvost J, Legrand J and Legentilhomme P 2001 Three-dimensional swirl flow velocity-field reconstruction using a neural network with radial basis functions J. Fluids Eng. 123 920–7

[38] Casa L and Krueger P 2013 Radial basis function interpolation of unstructured, three-dimensional, volumetric particle tracking velocimetry data Meas. Sci. Technol. 24 065304

[39] Al Mamun S, Lu C and Jayaraman B 2018 Extreme learning machines as encoders for sparse reconstruction Fluids 3 88

[40] 1997 BS8488, Fans for general purposes, part 1: methods for testing performances (London: British Standard Institution)

[41] Akbari G and Montazerin N 2013 On the role of anisotropic turbomachinery flow structures in inter-scale turbulence energy flux as deduced from SPIV measurements J. Turbul. 14 44–70

[42] Najiari M R, Montazerin N and Akbari G 2015 On the presence of spectral shortcut in the energy budget of an asymmetric jet–wake flow in a forward-curved centrifugal turbomachinery as deduced from SPIV measurements J. Turbul. 16 503–24

[43] Akbari G, Montazerin N and Akbarizadeh M 2012 Stereoscopic particle image velocimetry of the flow field in the rotor exit region of a forward-blade centrifugal turbomachine Proc. Inst. Mech. Eng. A 226 163–81

[44] 2002 FlowManager software and introduction to PIV instrumentation, dantec dynamics A/S (Tonsbakken 18, DK-2740 Skovlunde, Denmark)

[45] Najiari M, Montazerin N and Akbari G 2012 Statistical PIV data validity for enhancement of velocity driven parameters in turbomachinery jet-wake flow 20th Annual Int. Conf. on Mechanical Engineering (Shiraz, Iran) pp 16–18

[46] Keane R D and Adrian R J 1992 Theory of cross-correlation analysis of PIV images Appl. Sci. Res. 49 191–215

[47] Liu S, Meneveau C and Katz J 1994 On the properties of similarity subgrid-scale models as deduced from measurements in a turbulent jet J. Fluid Mech. 275 83–119

[48] Pope S B 2000 Turbulent Flows (Cambridge: Cambridge University Press)

[49] Gonzalez O and Stuart A M 2008 A First Course in Continuum Mechanics (Cambridge: Cambridge University Press)

[50] Vapnik V, Golowich S E and Smola A J 1996 Support vector method for function approximation, regression estimation and signal processing Adv. Neural Inf. Process. Syst. 9 281–7

[51] Yu P-S, Chen S-T and Chang I-F 2006 Support vector regression for real-time flood stage forecasting J. Hydrol. 328 704–16

[52] Bergstra J and Bengio Y 2012 Random search for hyper-parameter optimization J. Mach. Learn. Res. 13 281–305

[53] Fumo N and Biswas M R 2015 Regression analysis for prediction of residential energy consumption Renew. Sustain. Energy Rev. 47 332–43

[54] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R and Dubourg V 2011 Scikit-learn: machine learning in Python Int. J. Mach. Learn. 12 2825–30
[55] Varma A K and Mitra D 2019 Statistical feature-based SVM wideband sensing IEEE Commun. Lett. 24 581–4
[56] Murray N E and Ukeiley L S 2007 An application of gappy POD Exp. Fluids 42 79–91
[57] Parker J A, Kenyon R V and Troxel D E 1983 Comparison of interpolating methods for image resampling IEEE Trans. Med. Imaging 2 31–39
[58] Lyra G B, Correia T P, De Oliveira-júnior J F and Zeri M 2018 Evaluation of methods of spatial interpolation for monthly rainfall data over the state of Rio de Janeiro, Brazil Theor. Appl. Climatol. 134 955–65
[59] Zhuang Y, Chen L, Wang X S and Lian J 2007 A weighted moving average-based approach for cleaning sensor data 27th Int. Conf. on Distributed Computing Systems (IEEE) pp 38–45
[60] Gonzalez R C and Woods R E 2018 Digital Image Processing 4th edn (New York: Pearson Education)
[61] Gunes H, Sirisup S and Karniadakis G E 2006 Gappy data: to Krig or not to Krig? J. Comput. Phys. 212 358–82
[62] Sirovich L 1987 Turbulence and the dynamics of coherent structures. Part I: coherent structures Q. Appl. Math. 45 561–71