33RD CONGRESS
OF THE INTERNATIONAL COUNCIL
OF THE AERONAUTICAL SCIENCES
STOCKHOLM, SWEDEN, 4-9 SEPTEMBER, 2022

ICAS 2022 SWEDEN

WING BUFFET PRESSURE LOAD PREDICTION BASED ON A HYBRID DEEP LEARNING MODEL

Rebecca Zahn[1], Andre Weiner[2] & Christian Breitsamter[3]

[1]Technical University of Munich
[2]Technical University of Braunschweig
[3]Technical University of Munich

Abstract

In the present study, a hybrid deep learning reduced order model (ROM) is applied for unsteady transonic wing buffet load prediction. The hybrid model is defined by the combination of a convolutional variational autoencoder (CNN-VAR-AE) and a long short-term memory (LSTM) neural network. In the first step, the CNN-VAR-AE is trained using experimental buffet data. Thereby, the high-dimensional buffet flow field is reduced into a low-dimensional latent space. In the second step, the LSTM is trained and applied in order to predict the temporal evolution of the wing buffet pressure loads. As a test case, the generic XRF-1 configuration developed by Airbus, is applied. The XRF-1 configuration has been investigated at different transonic buffet conditions in the European Transonic Wind Tunnel (ETW). During the test campaign, surface pressure data has been obtained by means of unsteady pressure sensitive paint (iPSP) measurements. As a first step, the trained model is applied in a recurrent multi-step prediction mode in order to reproduce pressure distribution at flow conditions included in the validation data set. In the second step, the trained model is used for the prediction of pressure distributions at an unknown flow condition. A comparison of the experimental data with data predicted by the deep learning model yields an accurate prediction of the buffet flow characteristics.

Keywords: Deep Learning, Convolutional Autoencoder, Long Short-Term Memory Neural Network, Transonic Wing Buffet Aerodynamics

1. Introduction and Motivation

Transonic wing buffet, also referred to as wing shock buffet, represents an unsteady aerodynamic phenomenon which occurs during certain flow conditions in the transonic regime. The self-sustained instability of the buffet phenomenon is characterized by shock-wave oscillations and intermittent boundary layer separation, which leads to undesired vibrations of the aircraft structure. Therefore, the determination of buffet loads is of paramount importance for the safety and efficiency of a civil aircraft.

Nowadays, a variety of computational methods for the prediction of unsteady aerodynamic loads is available. For the investigation of transonic buffet flows, which are dominated by nonlinear flow features, high-fidelity Computational Fluid Dynamics (CFD) methods such as Unsteady Reynolds-Averaged Navier-Stokes (URANS) or Detached-Eddy Simulation (DES) are commonly applied [1, 2]. Further, experimental investigations have been used to some extend for the determination of unsteady buffet loads [3, 4]. However, although high-performance computing and experimental facilities are available, the accurate computation of unsteady buffet loads is challenging due to the high computational costs and time, especially if realistic aircraft configurations are considered.

An alternative to the application of high fidelity solutions is given by system identification methods. These methods are characterized by the representation of an unknown aerodynamic system, defined by a certain number of inputs and outputs. In unsteady aerodynamic modeling, the inputs are usually defined by the motion of the body, whereas the output denotes the corresponding, integrated and

local aerodynamic loads [5]. The unknown aerodynamic system can be treated as a black-box with a reduced number of degrees of freedom compared to the full-order approach. Therefore, these methods are commonly referred to as Reduced-Order Models (ROMs).

With focus on unsteady aerodynamic modeling, various ROM techniques such as Wiener models [6], Volterra series [7] and Kriging models [8] have been successfully applied. Further, recurrent neural networks (RNN) have been used by Mannarino and Mantegazza [9] for the prediction of limit cycle oscillations. Further, radial basis function neural networks (RBF-NN) have been employed by Zhang et al. [10] and Winter and Breitsamter [11] for flutter analysis and airfoil load prediction, respectively. In addition, Winter and Breitsamter [12] proposed a nonlinear ROM based on a series connection of a recurrent neuro-fuzzy model (NFM) and a multilayer perceptron (MLP) neural network for non-linear aerodynamic time series prediction.

However, considering the nonlinear ROM methods mentioned above, the prediction of spatio-temporal characteristics, such as unsteady pressure distributions, is not feasible due to the large number of output variables [13]. In order to solve this issue, several studies dealt with the application of dimensionality reduction techniques, such as Proper Orthogonal Decomposition (POD) and Dynamic Mode Decomposition (DMD). Park et al. [14] applied POD in combination with a neural network for wing design optimization tasks. Considering buffet aerodynamics, Timme [15] and Ohmichi et al. [16] used POD and DMD in order to identify instability mechanisms and dominant flow modes of three-dimensional wing buffet. Further, a study by Candon et al. [17] investigated DMD for dynamic bending and torsion load spectra prediction resulting from transonic buffet on a high-agility aircraft configuration.

Besides the application of dimensionality reduction techniques, deep learning methods such as convolutional neural networks (CNN) have been applied for the prediction of spatio-temporal characteristics of high-dimensional flow field data. Afshar et al. [18] applied a CNN for the prediction of the velocity and pressure field around an airfoil, depending on the shape of the airfoil and corresponding flow parameters. Sekar et al. [19] used a CNN for feature extraction of an airfoil and further processed them as an input for a MLP neural network in order to predict steady flow fields. In order to enable a better and faster prediction of temporal characteristics of flow field data, several studies proposed hybrid deep learning models based on CNN and RNN. Li et al. [20] applied a convolutional long short-term memory (ConvLSTM) neural network for the prediction of supersonic cascaded channel flow. Hasegawa et al. [21] proposed a series connected ROM based on a convolutional autoencoder (CNN-AE) and a LSTM for the prediction of unsteady-flow characteristics around bluff bodies with different shapes. Further, Nakamura et al. [22] applied the proposed CNN-AE/LSTM model for the prediction of turbulent channel flow.

Based on the aforementioned studies, the following study deals with the application of the hybrid CNN-AE/LSTM neural network as proposed by [21] for the prediction of wing buffet pressure loads. However, compared to previous studies, which considered numerical data sets, experimental flow field data is used for the training of the hybrid neural network. In particular, surface pressure data originated from unsteady pressure sensitive paint (iPSP) measurements, which have been conducted by the German Aerospace Center (DLR), is used. The experimental data is obtained during a wind tunnel test campaign in the European Transonic Wind tunnel (ETW), as part of the research project FOR2895, investigating high speed stall due to transonic buffet. As a test case, the Airbus XRF-1 configuration is investigated. During the measurement campaign, different flow conditions have been considered, covering buffet-onset and buffet conditions of the Airbus XRF-1 configuration. Based on the 3D buffet data, the deep learning model is trained. For performance evaluation, the trained model is applied for the prediction of buffet loads considering a flow condition which is not included in the training data set. A comparison of experimental and predicted data indicates that the trained ROM is able to reproduce the buffet characteristics to some extent.

2. Reduced Order Modeling

In the following section, the applied hybrid deep learning model is introduced. The first subsection covers a brief introduction of convolutional neural networks (CNN) as well as the working principle of autoencoders (AE). In the second part of the section, the LSTM neural network applied for time-series prediction, is discussed in detail.

2.1 Convolutional Neural Network (CNN)

A convolutional neural network (CNN), which has been proposed by LeCun et al. [23], is a type of neural network developed for processing spatial information. In general, a CNN is composed of three main layers: a convolutional layer, a pooling layer and a fully-connected (FC) layer. However, the convolutional layer represents the main building block of each CNN architecture. Considering a three-dimensional image defined by an input array X, the convolution operation performed in the convolution layer can be defined as follows:

$$Y_{i,j,k} = \sum_{l=0}^{C-1} \sum_{m=0}^{H-1} \sum_{n=0}^{W-1} X_{i+m,j+n,k} K_{mnlk} + b_k \tag{1}$$

with $X_{i,j,k}$ denoting the input at index (i,j,k), K_{mnlk} defining the weights at point (m,n,l) in the k-th filter and b_k represents the bias in the k-the filter. The output of the convolution layer is denoted as $Y_{i,j,k}$.
The weight matrix K associated with the multi-channel convolution has the size $C_{out} \times C_{in} \times H_w \times W_w$, whereas the size of the input X is defined by $C_{in} \times H_w \times W_w$. C_{out} denotes the number of feature maps in the channel dimension of output Y. The elements of the filter K and the bias b of the CNN are free parameters which are successively adapted during the training process.
The filter slides stepwise over the input data, performing element-wise multiplications with the elements of the input matrix. At each step, the results are summed up and written as a new output. In order to define the number of elements the filter selects at each step, a parameter referred to as *stride* (s) can be set. By applying the striding technique, the output size can be decreased compared to the input size. Choosing a stride of one, slides are picked an entry apart, resulting in an output size equal to the input size. If the stride is defined as two, the size of the input data is reduced by a factor of two and so on.
In the present work, the CNN architecture is used within an autoencoder structure, which is commonly referred to as a convolutional autoencoder (CNN-AE). The application of a CNN-AE aims for encoding a given input and reconstructing the input to a given output. The network structure is composed of three parts, the encoder, decoder and latent space. The encoder maps the high-dimensional data into a low-dimensional latent space and the decoder extends the data back into its original resolution. In order to learn the representation of the input data, the network is trained in an unsupervised way.
In order to enable a better training process, a convolutional variational AE (CNN-VAR-AE) is applied. Compared to a normal CNN-AE, the input is encoded as a normal distribution over the latent space, rather than a single point. From this distribution, a random point is selected, which is fed into the decoder. Based on the sampled point, the reconstruction error is computed and backpropagated through the CNN-VAR-AE.

2.2 Long Short-Term Memory (LSTM)

The LSTM architecture has been proposed by Hochreiter and Schmidhuber [24] and represents a type of recurrent neural network (RNN), which is capable of predicting both long- and short-term dependencies in time-series data. In contrast to a basic RNN, the application of the LSTM solves a problem known as vanishing gradient problem [25], in which the training performance of the network becomes insufficient after a small number of training iterations. Further, the representation of time-delayed effects, which are important for unsteady aerodynamic modeling, are incorporated in the LSTM architecture [5]. The LSTM cell processes the incoming information through three gates, namely the forget gate f, the input gate i and the output gate o. In Fig. 1, the architecture of a LSTM cell including the characteristic gates, is illustrated.

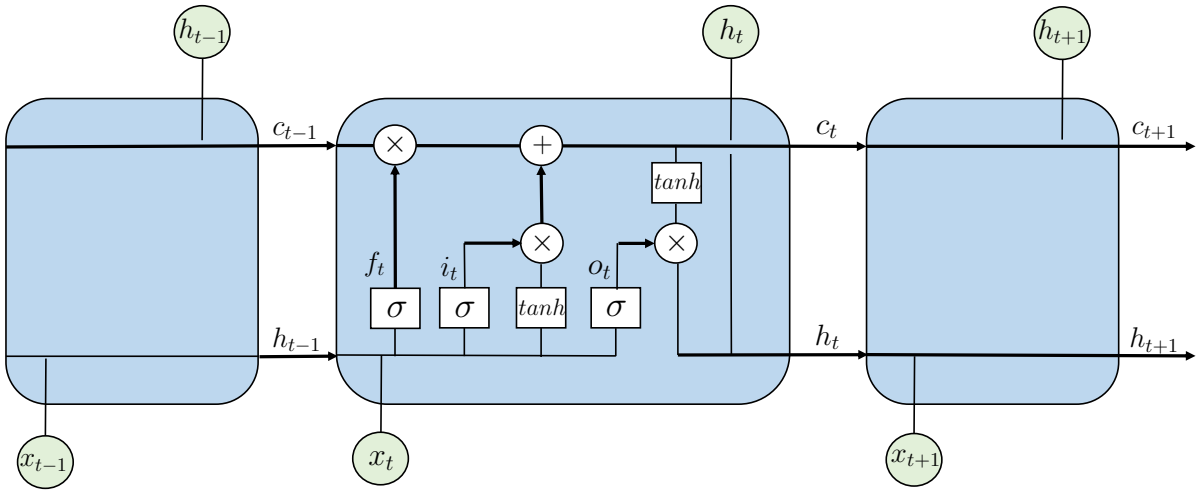

Figure 1 – Architecture of the LSTM cell including the forget (f), input (i) and output (o) gate.

The difference equations which define the forward pass, the cell state update and the output processing are summarized in the following:

$$\begin{aligned} f_t &= a(W_f x_t + W_f h_{t-1} + b_f) \\ i_t &= a(W_i x_t + W_i h_{t-1} + b_i) \\ \tilde{c}_t &= b(W_c x_t + W_c h_{t-1} + b_c) \\ c_t &= f_t c_{t-1} + i_t \tilde{c}_t \\ o_t &= a(W_o x_t + W_o h_{t-1} + b_o) \\ h_t &= o_t b(c_t) \end{aligned} \tag{2}$$

In Equation 2, W_f, W_i, W_c and W_o denote the weight matrices for the respective gates, while b_f, b_i, b_c and b_o represent bias vectors. The activation functions a and b are defined as sigmoid σ and $tanh$ in the present study.
By passing the forget gate f, the input is filtered and partially discarded from the cell. After the data is updated in the forget gate, the selected information is passed to the input gate i, which decides how much the current cell state c should be updated with new data $\tilde{c}$. In the output gate o, the updated information is passed to the next layer or the output layer.

3. Hybrid Model Architecture

In order to enable an accurate prediction of wing buffet pressure loads, the CNN-VAR-AE and the LSTM as introduced in Sec. 2.1 and Sec. 2.2 are combined in order to form a hybrid deep learning model. In Fig. 2, the architecture of the hybrid model is visualized.
Both the encoder and decoder of the CNN-VAR-AE are divided into several levels. At each level, a number of operations is performed on the input data, which is fed into the encoder. Defining the input of the CNN-VAR-AE, snapshots of pressure distribution $c_p(\tau, x, y, z)$ at each measured time step τ are combined into a two-dimensional array $X_p^{(\tau)}$ along the channel dimension. Therefore, the size of the input array is defined as $n \times H \times W$. H and W denote the height and width of each c_p snapshot, which are defined by powers of two.
As a first step, a multi-channel convolution with a kernel size of 2×2 and stride s=2 is performed. Following the convolution, the input is normalized based on the mean and variance of each training batch, which is referred to as batch normalization [27]. In the last step, an activation function is applied to each element of the incoming array. Within the scope of this work, a rectified linear unit (ReLU) is used.

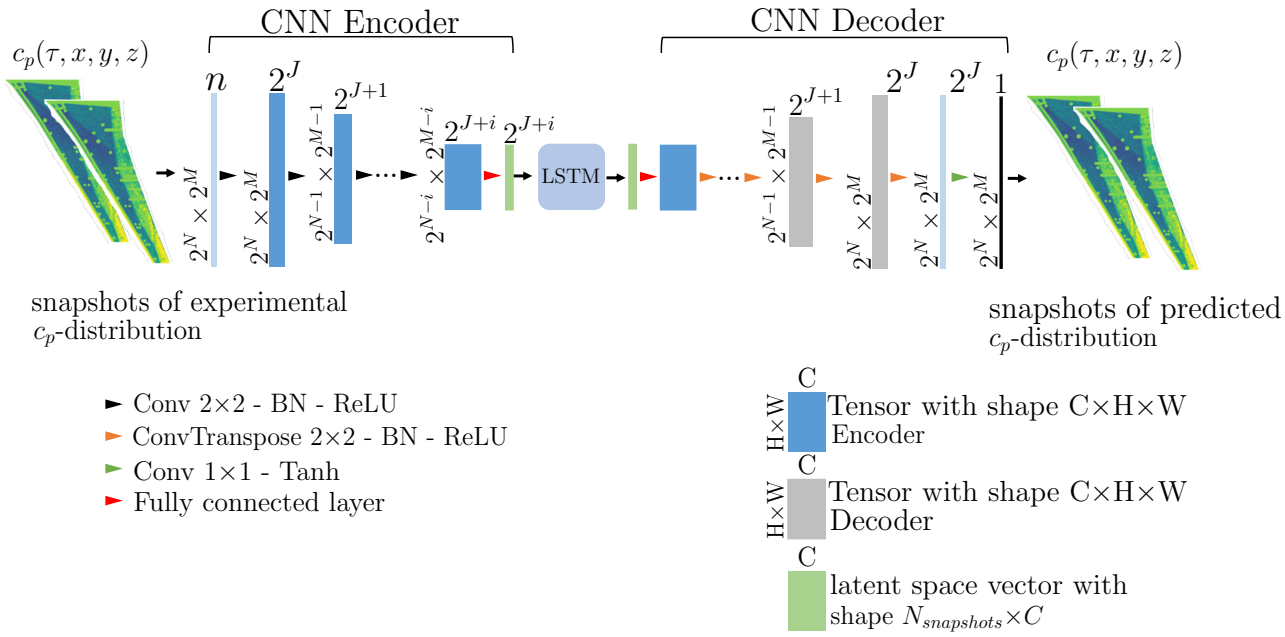

Figure 2 – Architecture of the hybrid deep learning model.

The sequence of operations described above is applied at each level of the encoder. Therefore, the output has twice as many channels as the input. However, the first convolution applied to the input array X increases the number of channels from n to C. In addition, the size of the spatial dimensions (H,W) is reduced by a factor of two at each level of the encoder. As the last encoder processing step, the data is fed into a FC layer in order to reduce the channel dimension. The resulting reduced latent features are then sampled from a normal distribution

After the input data has been processed by the encoder, the data is reshaped and fed into the LSTM neural network. The LSTM is used for the prediction of buffet pressure distributions at time step $k+m$ based on several previous snapshots of the pressure distribution at time steps $k-n+1$ to k. Here, n denotes the number of training samples applied for the prediction, whereas m defines the number of time steps ahead which should be predicted. In Fig. 3, the schematic of the multiple time steps ahead prediction obtained by the LSTM model, is depicted.

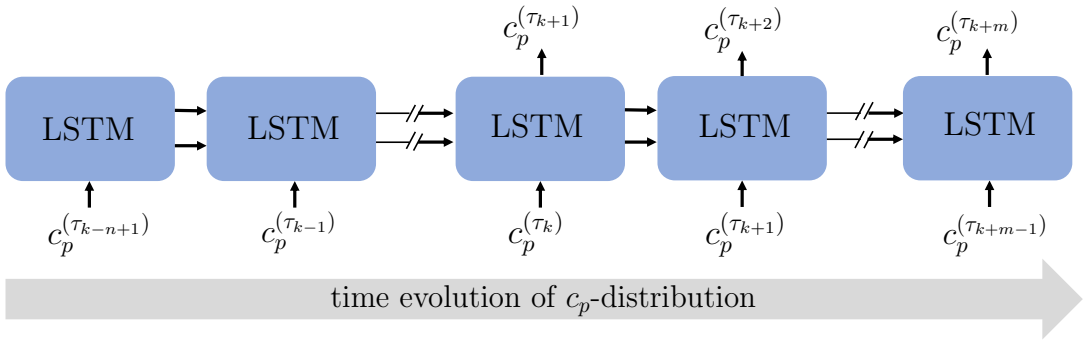

Figure 3 – Prediction of time evolution of the c_p-distribution as obtained by the LSTM.

As soon as the prediction of the LSTM is finalized, the output is fed into the decoder. Therefore, the low-dimensional data is reconstructed to the original high-dimensional flow field. At each level of the decoder, a transposed convolution is applied to upscale the spatial dimension. The corresponding kernel size and stride values are chosen equally to those of the encoder. In addition, batch normalization is applied at each level of the decoder. In order to obtain the final prediction, an activation function is applied to the output of the last decoder level. Therefore, a hyperbolic tangent $tanh$ is chosen, which reshapes all predicted elements to a value range of [-1,1].

4. Test Case: Airbus XRF-1

The experimental wing buffet pressure data used for training and application of the hybrid neural network has been obtained during a wind tunnel test campaign in the European Transonic Wind Tunnel (ETW). As a test case, the Airbus XRF-1 research configuration is applied. In Fig. 4, the wind tunnel model of the Airbus XRF-1 configuration is visualized.

Figure 4 – Front view of the Airbus XRF-1 wind tunnel model (©Airbus/ETW).

As depicted in Fig. 4, the model is equipped with a fixed vertical and horizontal tailplane as well as adjustable ailerons. In order to obtain the buffet pressure loads on the upper wing surface, optical measurements using unsteady pressure sensitive paint (iPSP) [28, 29] were conducted by the German Aerospace Center (DLR).
During the wind tunnel test campaign, different buffet conditions have been investigated, characterized by freestream Mach numbers of Ma_∞ = [0.84,0.87,0.9] and Reynolds numbers of Re = [3.3 Mio., 12.9 Mio., 25 Mio.]. In addition, for each flow condition, varying angles of attack α have been considered.

5. Application of the Hybrid Deep Learning Model

In the following section, the application process and the results of the hybrid neural network are discussed. In the first part of the section, the preprocessing steps of the experimental data are briefly described. In the second subsection, the selection of hyperparameters and the training procedure is outlined. In the last subsection, the result of the application of the trained model are presented in detail.

5.1 Data Preprocessing

In order to feed the experimental data as obtained by iPSP measurements into the deep learning model, the data needs to be preprocessed accordingly. All preprocessing steps are accomplished using the Python library flowTorch [30]. Based on the geometry of the XRF-1 wind tunnel model and the iPSP measurement technique, the pressure distribution on the upper wing surface is discretized by 465×159 data points.
Due to measurement errors during the wind tunnel measurement campaign, the data set includes a small amount of non physical c_p values. Therefore, in the first preprocessing step, the experimental data is cleaned by applying a weight mask, which defines values of $c_p \geq 1.5$ as 1 and values of $c_p \leq -1.5$ as 0.
In the second step, the number of data points representing the pressure distribution is downscaled by linear interpolation from 465×159 to 256×128 ($2^8 \times 2^7$). In Fig. 5, a comparison of the wing pressure distribution represented by the original amount of data points (left) and the interpolated data points (right), is visualized.

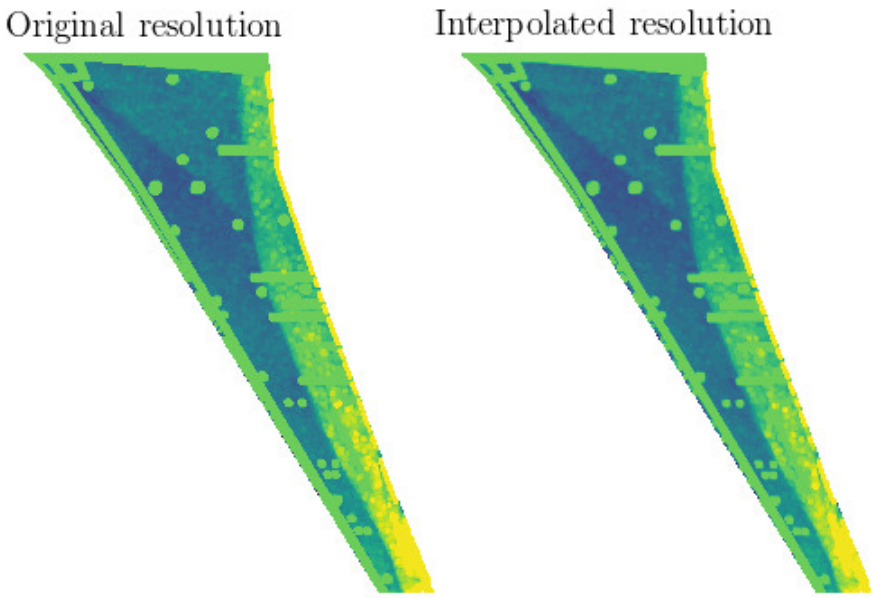

Figure 5 – Original resolution (left) and interpolated resolution (right) of the experimental data.

As shown in Fig. 5, the reduced resolution still maintains a high level of detail of the spatial resolution. Further, rescaling the surface resolution to an input array size of powers of two enables faster processing of the neural network.
In the final step, the data set is normalized based on the minimum and maximum pressure values in the data set ($c_{p,min}$,$c_{p,max}$). Therefore, the resulting value range is [-1,1].

5.2 Training of the Hybrid Model

The training procedure of the hybrid deep learning model includes two consecutive steps. In the first step, the CNN-VAR-AE is trained independently, using snapshots of flow conditions which should be predicted by the LSTM. In the second step, the LSTM is trained using the reduced snapshot data set as obtained by the CNN-VAR-AE encoder.
Since the application of the trained model aims for both a recurrent multi-step ahead prediction mode as well as a data interpolation mode, two different training procedures are considered. For the multi-step ahead prediction mode, the flow condition of interest is applied for training, whereas for the data interpolation mode various flow conditions are used. Considering the second application case, the flow conditions used for training define the value range for the interpolation of additional flow conditions. In the following, the focus is on the prediction of buffet characteristics considering a variation of the angle of attack.
As a first step, the CNN-VAR-AE is trained using data representing a single flow condition. Therefore, iPSP data measured at a flow condition of Ma_∞ = 0.9, Re = 25 Mio. and α = 4°, is applied. The CNN-VAR-AE is trained using 1000 c_p snapshots, whereas 200 snapshots are used for hyperparameter tuning and validation. During the training, the snapshots are fed into the encoder in batches, including 32 time steps each. For both the encoder and decoder five convolution levels are applied, which reduces the input array size of $64 \times 256 \times 128$ to $512 \times 32 \times 16$. By passing the FC layer, the channel size is downscaled from 512 to 256 features. The initial learning rate is set to 10^{-4} and the training of the CNN-VAR-AE is terminated after 1000 epochs.
Considering the second application case, the CNN-VAR-AE is trained based on snapshots originated from two different flow conditions, represented by Ma_∞ = 0.9, Re = 25 Mio. and α = [4°,6°]. For the training, 2000 c_p snapshots are applied, with 1000 snapshots representing each flow condition. For validation, in total 400 snapshots are used. The parameters are chosen equally to the training procedure mentioned above. In order to achieve an appropriate model performance, the mean squared error (MSE) (see Eq.(3)) between the reference experimental data and the predictions of the hybrid model, is minimized.

$$MSE = \frac{1}{N_S} \sum_{i=1}^{N_S} (\hat{y}_i - y_i)^2 \tag{3}$$

In Eq.(3), $\hat{y}$ and y represent the data as obtained by the neural network and the experiment, respectively. N_S equals the number of available snapshots. In Figure 6, training and validation losses of the CNN-VAR-AE for both training modes, are visualized. Therefore, the decrease in MSE (see Eq. 3) is plotted over the number of training iterations, also referred to as epochs.

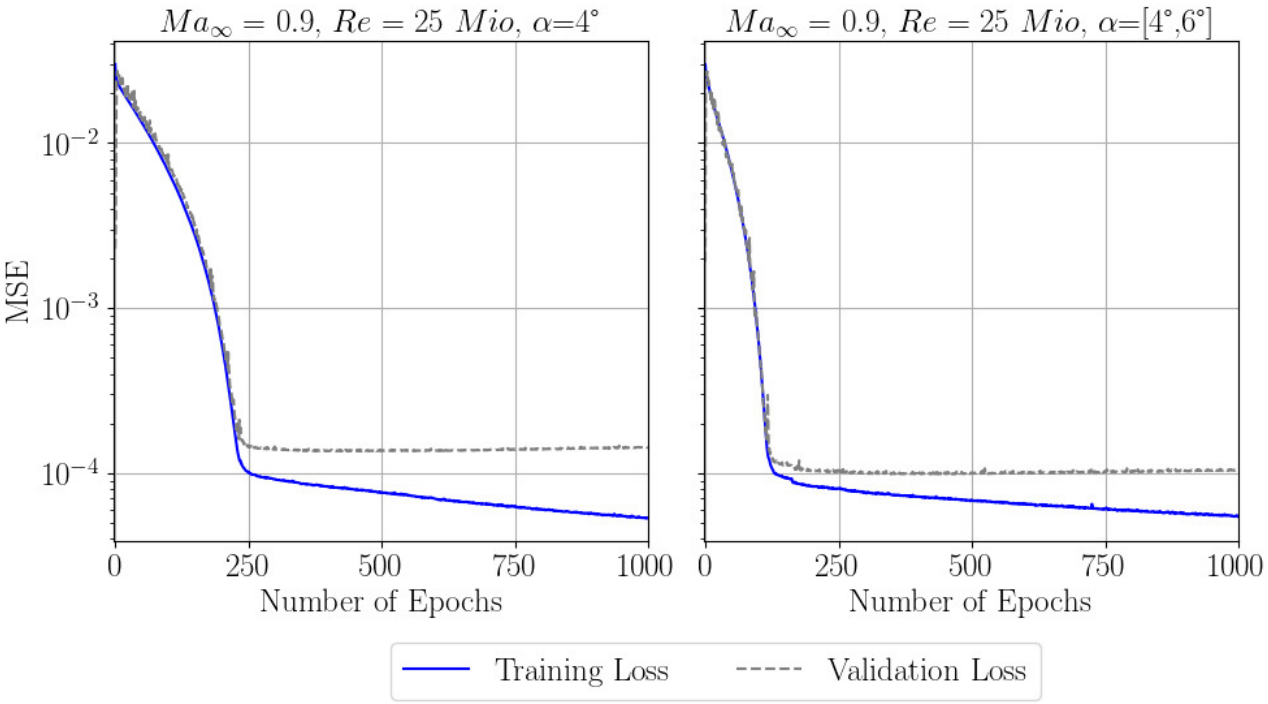

Figure 6 – Convergence trends of both training modes of the CNN-VAR-AE (single flow condition (left), multiple flow conditions (right)).

In addition to the corresponding convergence trends of training and validation losses, the training performance of the CNN-VAR-AE is evaluated based on a comparison of original validation snapshots and corresponding predictions. Therefore, one original snapshots and the corresponding snapshot predicted by the CNN-VAR-AE are exemplary visualized in Fig. 7 and Fig. 8, considering angles of attack of α = 4° and α = 6°, respectively. In addition, the MSE between the original and predicted snapshots is depicted in both figures on the left side.
Equally to the training of the CNN-VAR-AE, 1000 snapshots are used for training the LSTM. The overall amount of training snapshots is divided in sequences of 32 snapshots, while the batch size is defined as one. For the training, a stacked LSTM with four layers is applied, with each layer including 256 neurons. Analog to the training of the CNN-VAR-AE, the initial learning rate is defined as 10^{-4}. The LSTM is trained for 500 epochs, until a sufficient convergence is reached.

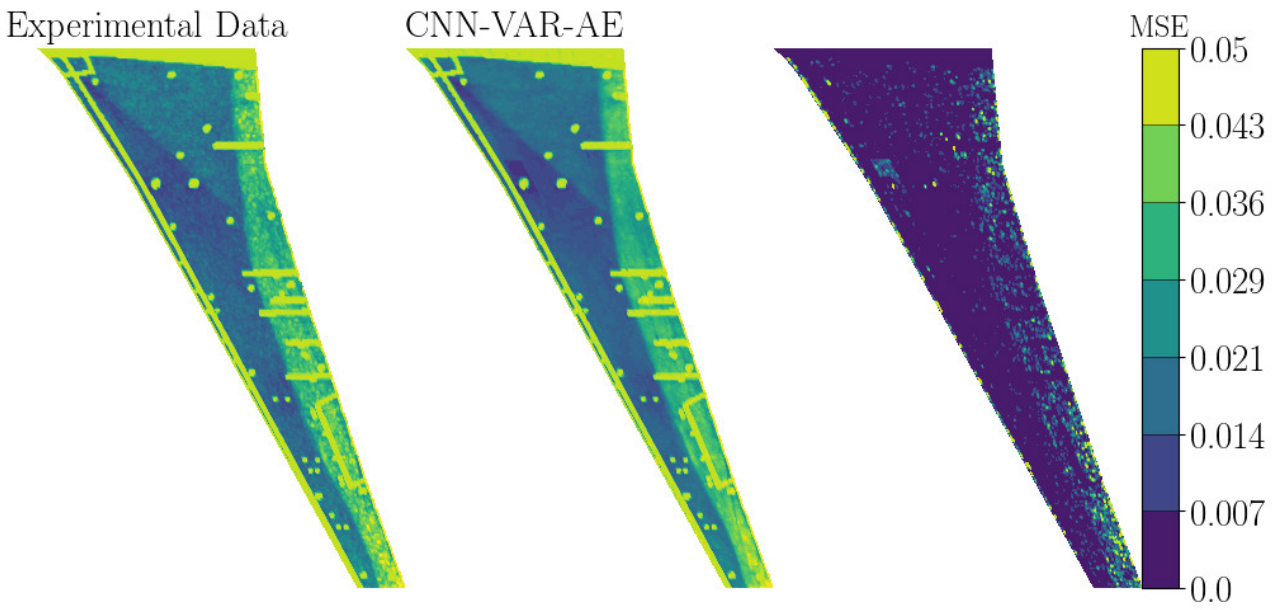

Figure 7 – Comparison of experimental (left) and predicted (middle) snapshots by the trained CNN-VAR-AE ($Ma_\infty = 0.9$, Re = 25 Mio., α = 4°). The corresponding MSE is additionally depicted on the right wing surface plot.

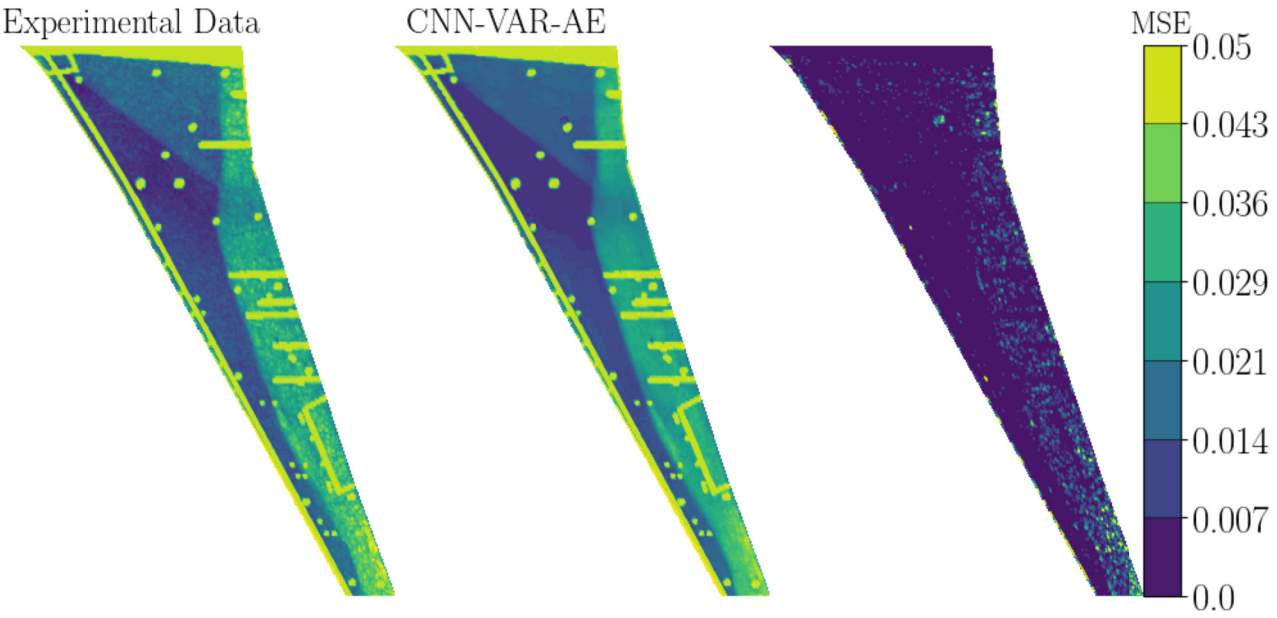

Figure 8 – Comparison of experimental (left) and predicted (middle) snapshots by the trained CNN-VAR-AE ($Ma_\infty = 0.9$, Re = 25 Mio., α = 6°). The corresponding MSE is additionally depicted on the right wing surface plot.

5.3 Investigation of Prediction Accuracy

5.3.1 Multi-step Predictions on the Validation Data Set

As a first step, the trained hybrid model is applied to the validation data set by performing recurrent multi-step predictions. Therefore, an initial number of snapshots included in the validation data set is presented to the model during the training process. By choosing the number of initial snapshots equal to the batch size (=32), the LSTM is applied for the prediction of 200 snapshots ahead. In Fig. 9 and Fig. 10, the results of the recurrent multi-step ahead prediction are visualized for two selected timesteps t = [40,86].

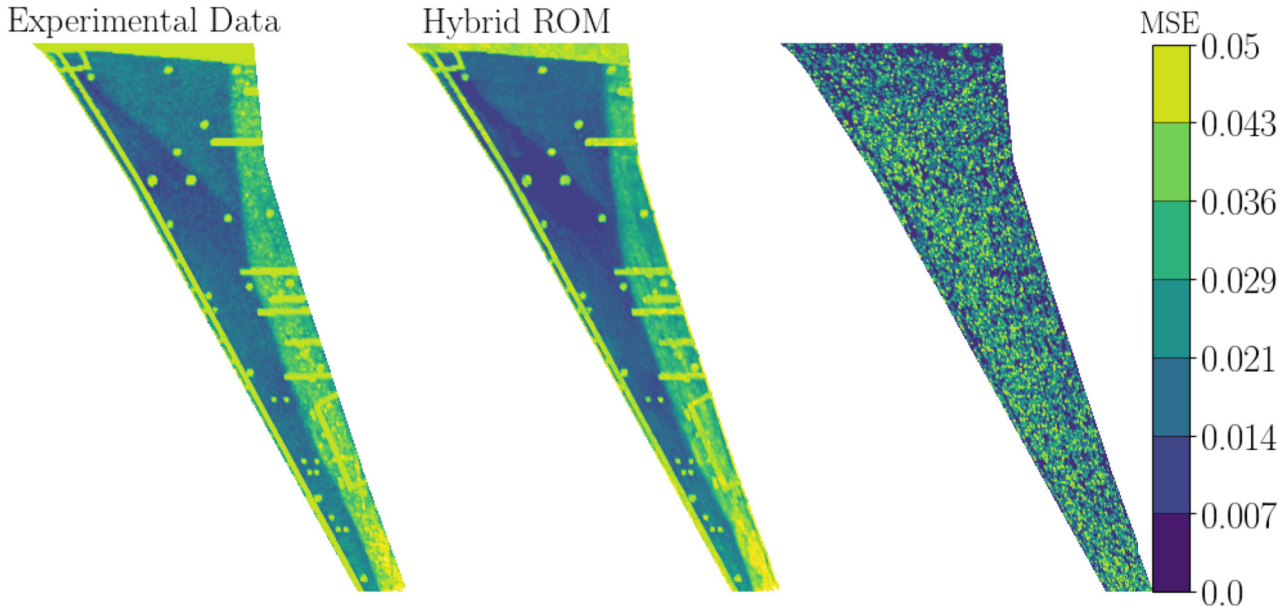

Figure 9 – Comparison of experimental (left) and predicted (right) pressure distribution at timestep t = 40. (Multi-step prediction mode, validation data set, Ma_∞ = 0.9, Re = 25 Mio., α = 4°).

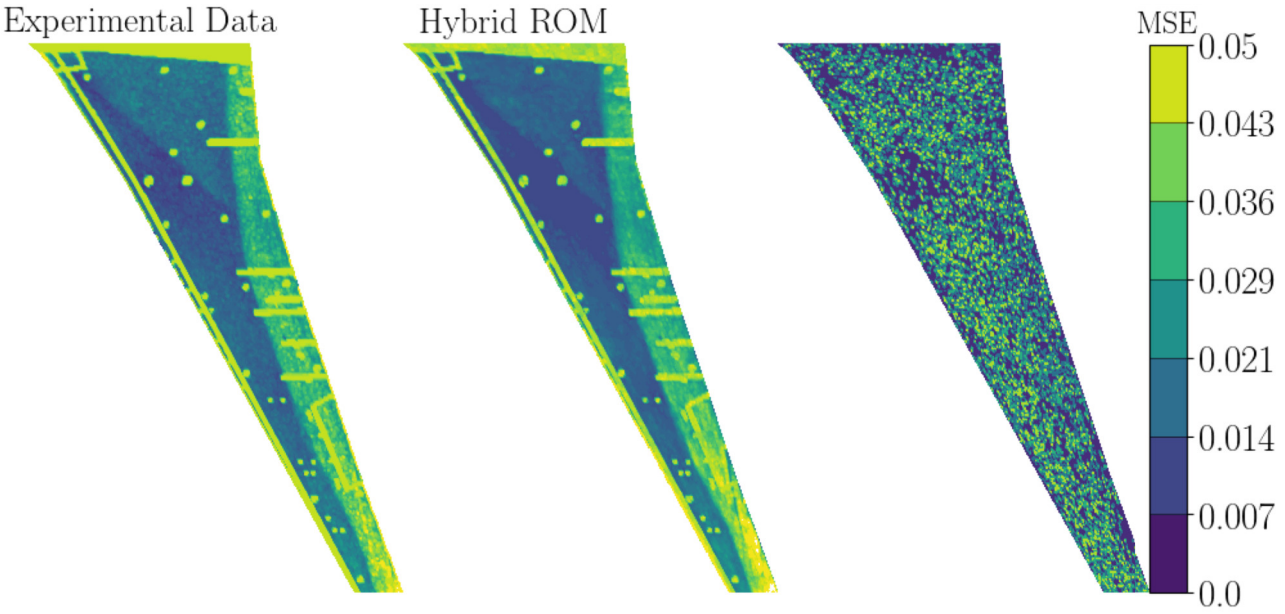

Figure 10 – Comparison of experimental (left) and predicted (right) pressure distribution at timestep t = 86. (Multi-step prediction mode, validation data set, Ma_∞ = 0.9, Re = 25 Mio., α = 4°).

Examining the distribution of the MSE between the experimental pressure distribution and the corresponding prediction obtained by the hybrid ROM, only small deviations in pressure distribution are indicated. However, in order to identify if the hybrid ROM is able to correctly reproduce the buffet flow physics, both experimental and predicted results are compared by means of modal analysis, in particular Single Value Decomposition (SVD). By applying SVD, important modes of the buffet instability are extracted, with the order of the modes yielding the contribution to the buffet flow. Therefore, the first six modes identified by SVD are considered in the following, while higher modes are neglected. In Fig. 11, the resulting frequency spectra of the SVD modes are visualized, yielding an overall good agreement.

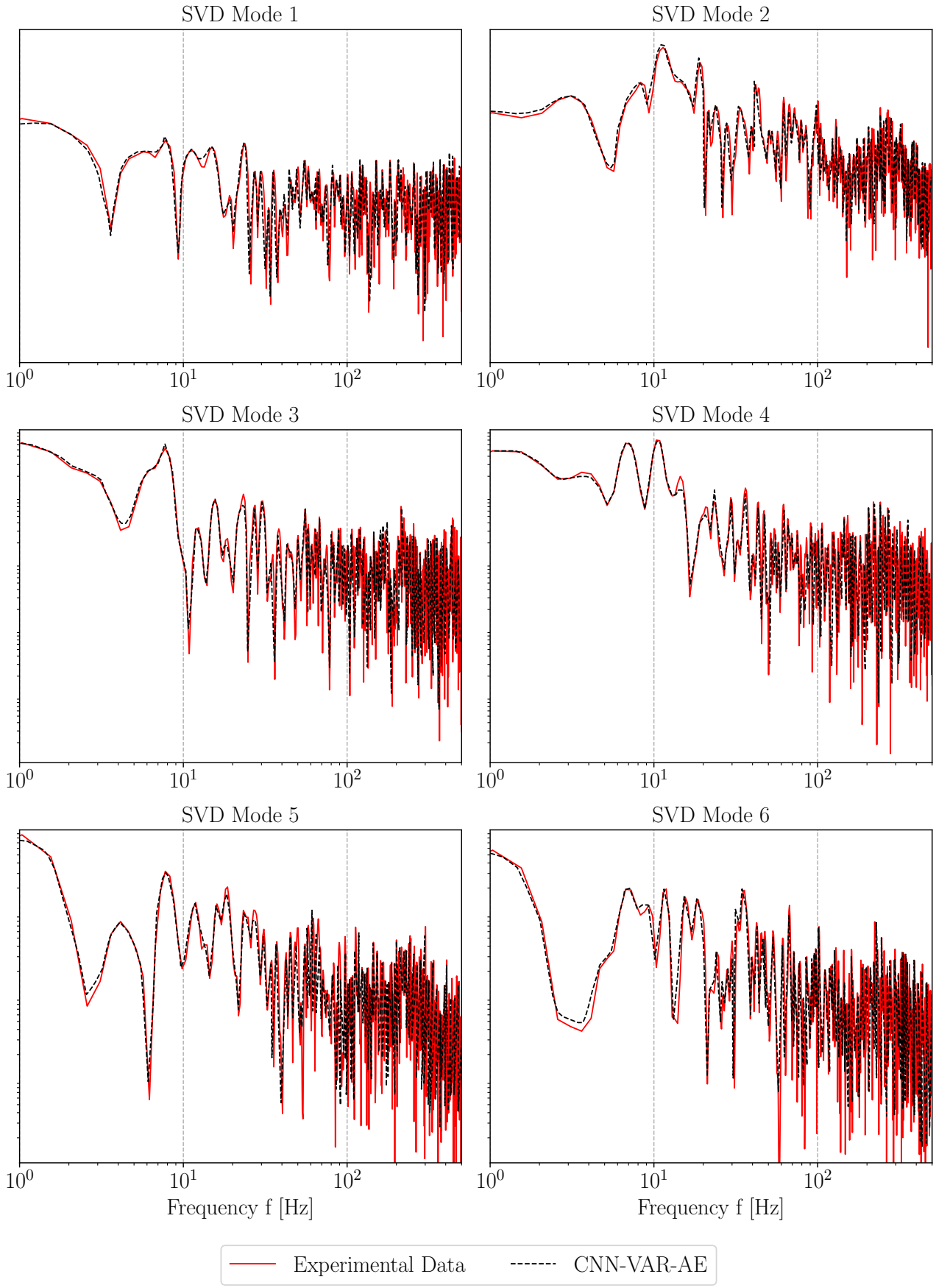

Figure 11 – Frequency domain responses of the first 6 modes of the buffet cycle ($Ma_\infty = 0.9$, Re = 25 Mio., α = 4°). The experimental data are compared to the data obtained by multi-step predictions.

5.3.2 Interpolation of Pressure Distributions

As a final step, the trained hybrid ROM is used for the prediction of pressure distributions at flow conditions which are not included in the training and validation data set. Therefore, in a first step, the trained CNN-VAR-AE is applied for data prediction. Since the c_p snapshots included in the training data set included angle of attacks of α = [4°,6°], the flow condition for testing the CNN-VAR-AE is chosen as $Ma_\infty = 0.9$, Re = 25 Mio. and α = 5°. In Fig. 12, a comparison between an original and predicted snapshot by the CNN-VAR-AE is visualized.

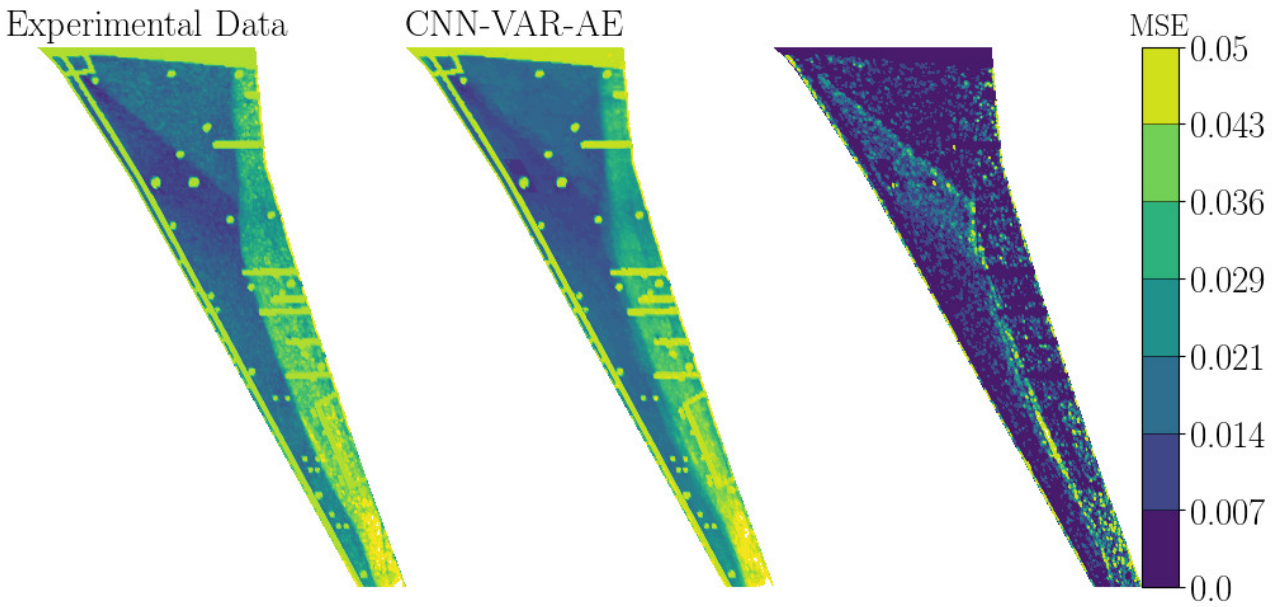

Figure 12 – Comparison of original snapshots and snapshots interpolated by the trained CNN-VAR-AE ($Ma_\infty = 0.9$, Re = 25 Mio., α = 5°).

Examining the c_p snapshots as interpolated by the CNN-VAR-AE, an overall good prediction is indicated. However, larger MSE values yield slight disagreement along the position of the shock. However, it has to be emphasized that the focus of the prediction is on the representation of the buffet flow physics such as shock movement and separation intensity, rather than the correct representation of the shock position on the wing.

In order to verify the dynamics of the predicted c_p snapshots, the resulting modes of the shock oscillation are also compared using SVD. In Fig. 15, the spectra of the first six modes are visualized. Comparing the mode spectra of the experimental data and the data obtained by the trained CNN-VAR-AE, an overall accurate agreement is indicated.

Finalizing the validation of the trained CNN-VAR-AE, the flow condition of interest is fed into the encoder and passed through the trained LSTM. Here, the LSTM is applied in a multi-step prediction mode using the unknown test data set. In Fig. 13 and Fig. 14, a comparison between the experimental snapshots and the predicted snapshots obtained by the LSTM, is visualized. Here, two timesteps t = [30,60] are considered. In addition, the corresponding spectra of the SVD modes are depicted in Fig. 16, yielding a precise prediction performance.

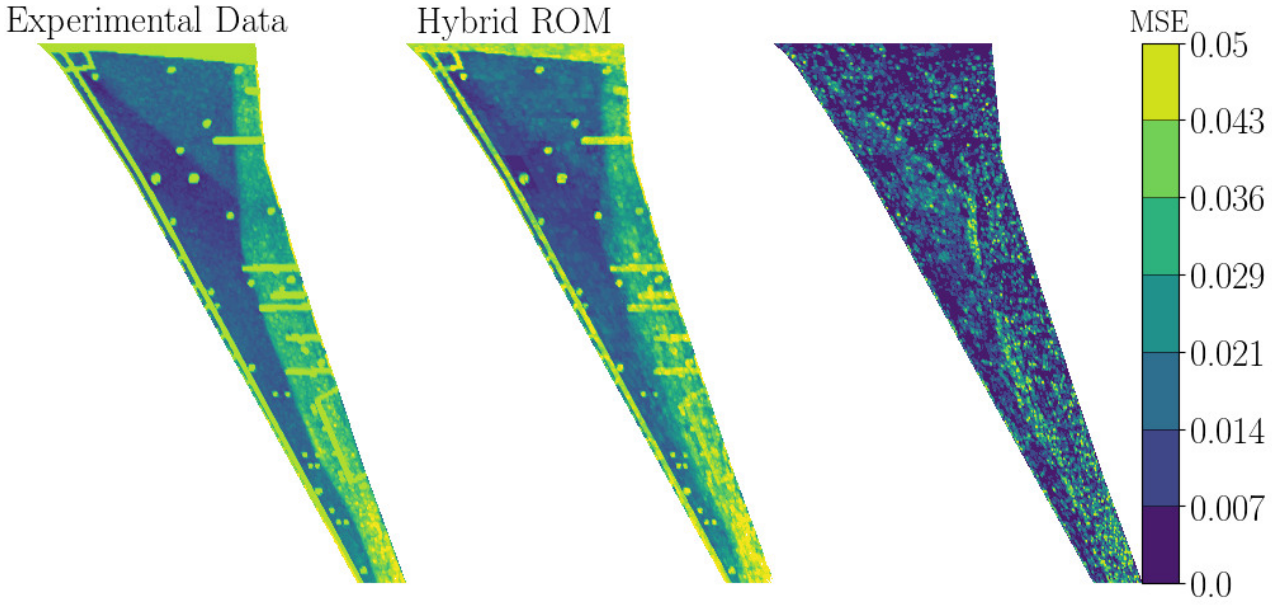

Figure 13 – Comparison of experimental and interpolated pressure distribution at timestep t = 30 ($Ma_\infty = 0.9$, Re = 25 Mio., α = 5°).

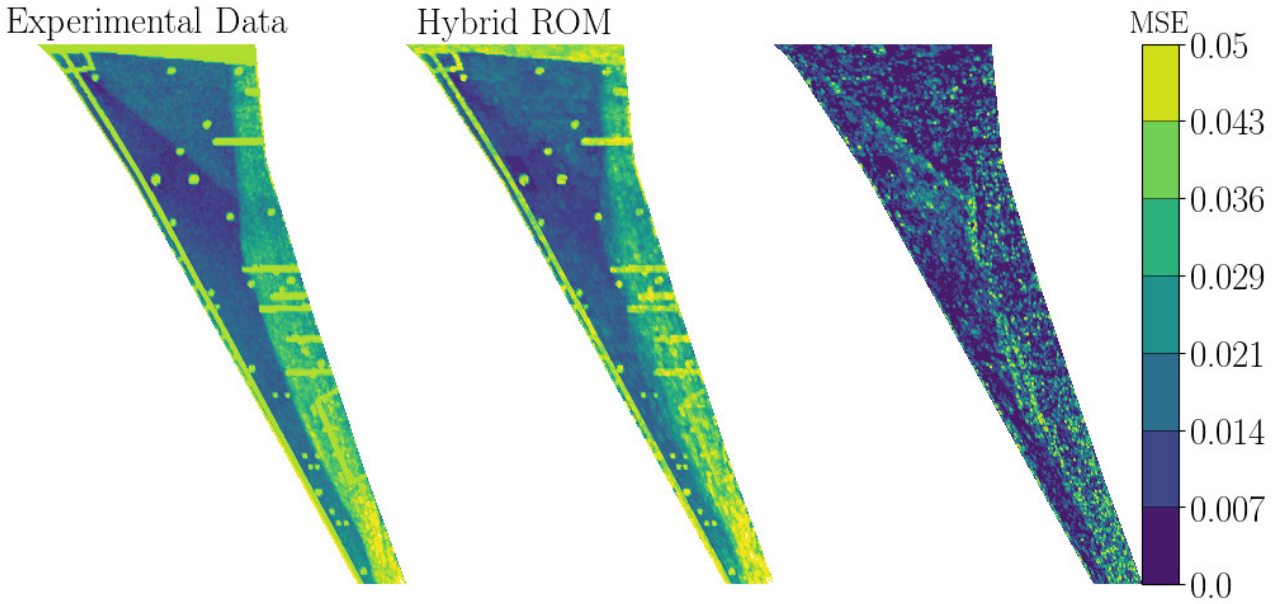

Figure 14 – Comparison of experimental and interpolated pressure distribution at timestep t = 60 ($Ma_\infty = 0.9$, Re = 25 Mio., α = 5°).

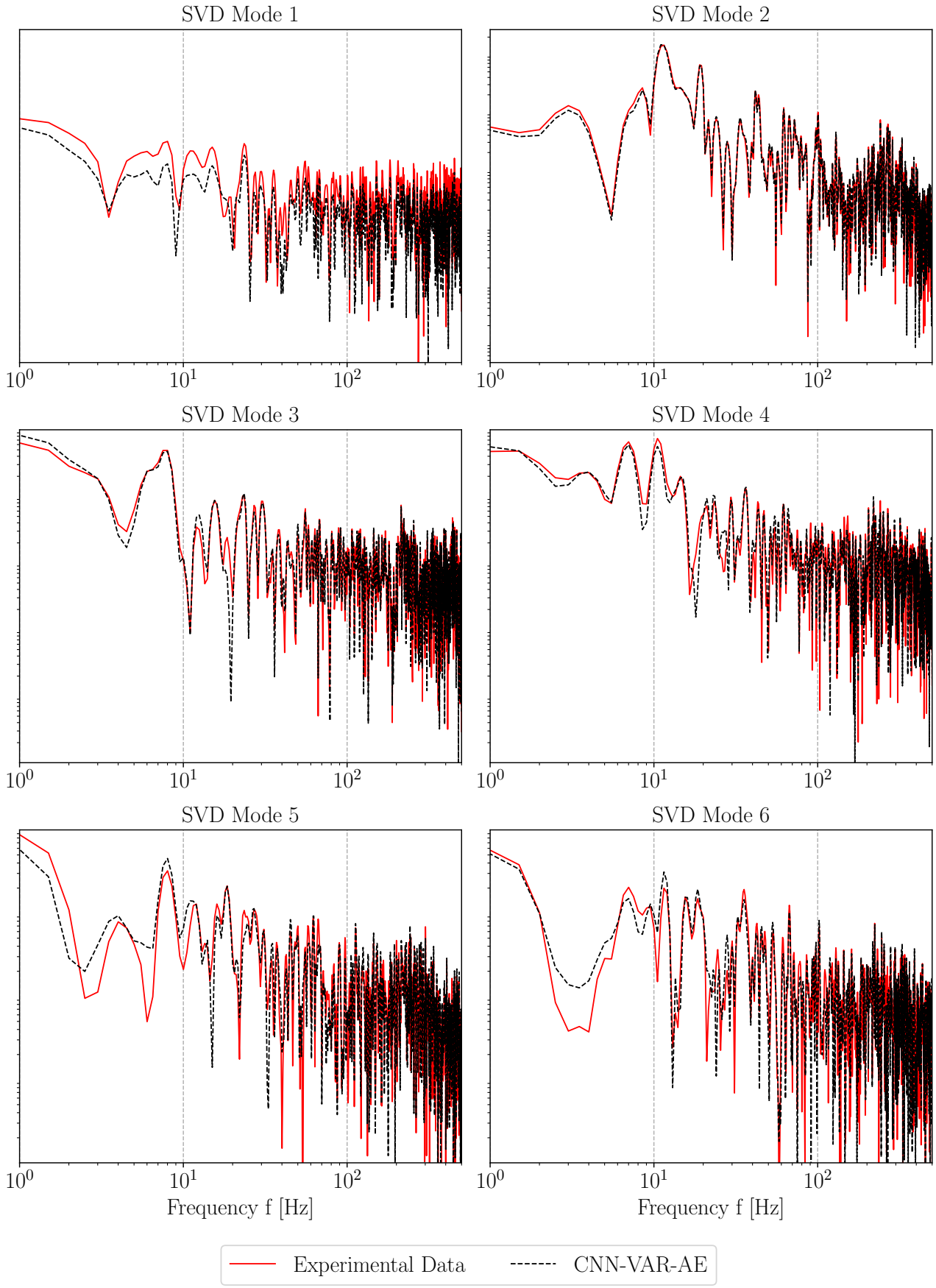

Figure 15 – Frequency domain responses of the first 6 modes of the buffet cycle ($Ma_\infty = 0.9$, Re = 25 Mio., α = 5°). The experimental results are compared to the results predicted by the CNN-VAR-AE.

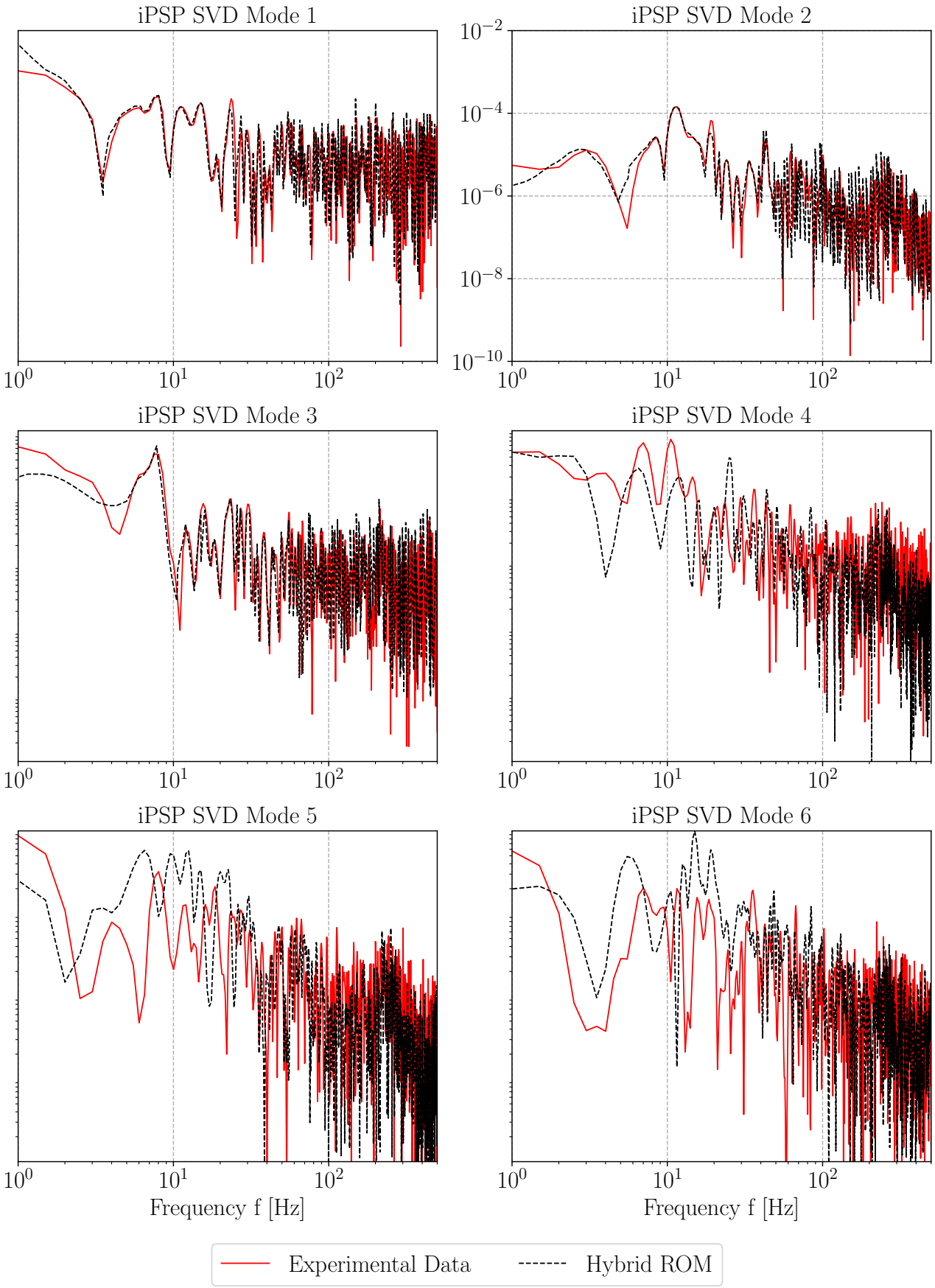

Figure 16 – Frequency domain responses of the first 6 modes of the buffet cycle ($Ma_\infty = 0.9$, Re = 25 Mio., α = 5°). The experimental results are compared to the results predicted by the hybrid ROM.

6. Conclusion

In the present study, a hybrid deep learning model based on a convolutional variational autoencoder (CNN-VAR-AE) and a long short-term memory (LSTM) neural network has been applied for the prediction of wing pressure distributions originated from transonic buffet. For training the neural network, experimental data as obtained by unsteady pressure sensitive paint (iPSP) measurements in the European Transonic Windtunnel (ETW), has been selected. As a test case, the Airbus XRF-1 configuration has been applied. After finalizing the training of the hybrid model, two application modes are presented. In the first step, the trained model is used in a recurrent multi-step prediction mode, reproducing the data used for neural network validation. In the second step, the trained model is applied for the prediction of buffet pressure distributions at unknown flow conditions. The results obtained by the trained model are compared to experimental data by means of a mode decomposition method. Examining the results, a precise prediction for both application scenarios is indicated.

Acknowledgements

The authors gratefully acknowledge the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for funding this work in the framework of the research unit FOR 2895 (Unsteady flow and interaction phenomena at high speed stall conditions), subproject TP7, grant number BR1511/14-1. Further, the authors would like to thank the Helmholtz Gemeinschaft HGF (Helmholtz Association), Deutsches Zentrum für Luft - und Raumfahrt DLR (German Aerospace Center) and Airbus for providing the wind tunnel model and financing the wind tunnel measurements.

Contact Author Email Address

rebecca.zahn@tum.de

Copyright Statement

References

[1] Sator, F., Timme, S.: Delayed Detached-Eddy Simulation of Shock-Buffet on Half Wing-Body Configuration, AIAA Journal, 55(4), pp. 1230-1240 (2016).

[2] Iovnovich, M., Raveh, D.E.: Numerical Study of Shock Buffet on Three-Dimensional Wings, AIAA Journal, 53(2), pp. 449-463 (2014).

[3] Paladini, E., Dandois, J., Sipp, D., Robinet, J.C.: Analysis and Comparison of Transonic Buffet Phenomenon over Several Three-Dimensional Wings, AIAA Journal, 57(1), pp. 1-18 (2016).

[4] Koike, S., Ueno, M., Nakakita, K., Hashimoto, A.: Unsteady Pressure Measurements of Transonic Buffet on the NASA Common Research Model, 4th AIAA Aplied Aerodynamic Conference, AIAA-2016-4044, Washington, D.C., (2016).

[5] Kou, J., Zhang, W.: Data-Driven Modeling for Unsteady Aerodynamics and Aeroelasticity, Progress in Aerospace Science, 125, pp. 110725 (2021).

[6] Kou, J., Zhang, W., Yin, M.: Novel Wiener Models with a Time-Delayed Nonlinear Block and Their Identification, Nonlinear Dynamics, 85(4), pp. 2389-2404 (2016).

[7] Silva, W.A.: Application of Nonlinear Systems Theory to Transonic Unsteady Aerodynamic Responses, Journal of Aircraft, 30(5), pp. 660-668 (1993).

[8] Glaz, B., Liu, L., Friedmann, P.P.: Reduced-Order Nonlinear Unsteady Aerodynamic Modeling Using a Surrogate-Based Recurrence Framework, AIAA Journal, 48(10), pp. 2418-2429 (2010).

[9] Mannarino, A., Mantegazza, P.: Nonlinear Aeroelastic Reduced Order Modeling by Recurrent Neural Networks, Journal of Fluids and Structures, 48, pp. 103-121 (2014).

[10] Zhang, W., Kou, J., Wang, Z.: Nonlinear Aerodynamic Reduced-Order Model for Limit-Cycle Oscillation and Flutter , AIAA Journal, 54(10), pp. 3304-3311 (2016).

[11] Winter, M., Breitsamter, C.: Reduced-Order Modeling of Unsteady Aerodynamic Loads Using Radial Basis Function Neural Networks, In: Deutscher Luft- und Raumfahrtkongress, Bonn, (2014).

[12] Winter, M., Breitsamter, C.: Nonlinear Identification via Connected Neural Networks for Unsteady Aerodynamic Analysis , Aerospace Science and Technology, 77, pp. 802-818 (2018).

[13] Rozov, V., Breitsamter, C.: Data-Driven Prediction of Unsteady Pressure Distributions Based on Deep Learning, Journal of Fluids and Structures, 104, (2021).

[14] Park, K.H., Jun, S.O., Baek, S.M., Cho, M.H., Yee, K.J., Lee, D.H.: Reduced-Order Model with an Artificial Neural Network for Aerostructural Design Optimization, Journal of Aircraft, 50(4), pp. 1106-1116 (2013).

[15] Timme, S.: Global Instability of Wing Shock-Buffet Onset, Journal of Fluid Mechanics, 885 (2020).

[16] Ohmichi, Y., Ishida, T., Hashimoto, A..: Modal Decomposition Analysis of Three-Dimensional Transonic Buffet Phenomenon on a Swept Wing, AIAA Journal, 56(10), pp. 3938-3950 (2018).

[17] Candon, M., Levinski, O., Altaf, A., Carrese, R., Marzocca, P.: Aircraft Transonic Buffet Load Prediction using Artificial Neural Networks, 59rd AIAA Structures, Structural Dynamics and Materials Conference, AIAA-2019-0763, San Diego, CA, (2019).

[18] Afshar, Y., Bhatnagar, S., Pan, S., Duraisamy, K., Kaushik, S.: Prediction of Aerodynamic Flow Fields Using Convolutional Neural Networks, Computational Mechanics, 64, pp. 525-545 (2019).

[19] Sekar, V., Jiang, Q., Shu, C., Khoo, B.C.: Fast Flow Field Prediction over Airfoils Using Deep Learning Approach, Physics of Fluids, 31, 057103 (2019).

[20] Li, Y., Chang, J., Wang, Z., Kong, C.: An Efficient Deep Learning Framework to Reconstruct the Flow Field Sequences of the Supersonic Cascade Channel, Physics of Fluids, 33, 056106 (2021).

[21] Hasegawa, K., Fukami, K., Murata, T., Fukagata, K.: Machine-Learning Based Reduced-Order Modeling of Flows Around Two-Dimensional Bluff Bodies of Various Shapes, Theoretical and Computational Fluid Dynamics, 34, pp. 367-388 (2020).

[22] Nakamura, T., Fukami, K., Hasegawa, K., Nabae, Y., Fukagata, K.: Convolutional Neural Network and Long Short-Term Memory Based Reduced Order Surrogate for Minimal Turbulent Channel Flow, Physics of Fluids, 33, 025116 (2021).

[23] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, 86(11), pp. 2278-2324 (1998).

[24] Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory, Neural Computation, 9(8), pp. 1735-1780 (1997).

[25] Hochreiter, S., Bengio, Y., Fransconi, P., Schmidhuber, J.: Gradient Flow in Recurrent Nets: The Difficulty of Learning Long-Term Dependencies, A Field Guide to Dynamical Recurrent Neural Networks, (2003).

[26] Kingma, D.P., Ba, J.L.: Adam: A Method for Stochastic Optimization, International Conference on Learning Representations (2015).

[27] Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating Deep Network Training By Reducing Internal Covariate Shift, Proceedings of the 32nd International Conference on Machine Learning, 37, pp. 448-456 (2015).

[28] Yorita, D., Klein, C., Henne, U., Ondrus, V., Beifuss, U., Hensch, A.-K., Longo, R., Guntermann, P., Quest, J..: Successful Application of Cryogenic Pressure Senstiive Paint Technique at ETW, 55th AIAA Aerospace Sciences Meeting, AIAA SciTech, Kissimee, FL, (2018).

[29] Liu, T., Sullivan, J.P., Asai, K., Klein, C., Egami, Y.: Pressure and Temperature Sensitive Paints, Second Edition (Experimental Fluid Mechanics), Chapter 9, Springer, Berlin, Germany (2021).

[30] Weiner, A., Semaan, R..: flowTorch - a Python library for analysis and reduced-order modeling of fluid flows, Journal of Open Source Software, 6(68), pp. 3860 (2021).