ADVANCED INTELLIGENT SYSTEMS
Open Access
www.advintellsyst.com

An Autoencoder-Based Deep-Learning Method for Augmenting the Sensing Capability of Piezoelectric Microelectromechanical System Sensors in a Fluid-Dynamic System

Mohammadrahim Kazemzadeh, Iman Mehdipour, Massimo De Vittorio, and Ferruccio Pisanello*

Herein, an innovative deep-learning architecture is proposed to enhance the sensing capabilities of a microelectromechanical system (MEMS) used in fluid dynamic applications. The MEMS sensor comprises a polyvinylidene fluoride flexible (PVDF) piezoelectric flag and a bluff body, with vortex generation influenced not only by the bluff body's geometry but also by the input fluid speed. As a result, mechanical vibrations are induced in the piezoelectric flag, leading to charge displacement and the generation of electrical voltage signals. Through the developed deep learning method, accurate extraction of wind speed and successful classification of turbulence are achieved. Experimental tests in a wind tunnel, involving various wind speeds and bluff body geometries, demonstrate the robust correlation between the extracted continuous manifold in Fourier spectra and wind speed. By incorporating a feed-forward network alongside the autoencoder, wind speed information even under strong turbulence is extracted. Moreover, the deep learning method's ability to classify different bluff bodies, independent of wind speed, is investigated. The findings reveal a unique capability to fingerprint turbulence and distinguish them for various applications. This research showcases the potential of our deep learning-based MEMS systems for enhancing fluid dynamic sensing and classification tasks.

1. Introduction

The dynamic response of a structure interacting with fluid flow (crosswind response) is referred to as fluid-induced vibration (FIV). Utilizing flexible microelectrcomechanical systems (MEMS transducers) assembled with a bluff body can be an efficient way of converting the kinetic energy of fluid flow into electrical energy to study the phenomena behind the bluff body shapes.[1] The shape of the bluff body splits the incoming flow into separate streams, leading to the creation of vorticity and turbulence. As a result, vibrations and flapping are experienced by the bluff body and transducer assembly when inserted into the fluid flow. The efficiency and effectiveness of the energy conversion is determined by a combination of factors including bluff body-induced drag and lift coefficients, vorticity pattern, flag configuration, and the fluid flow regime.[2]

Crosswind responses may occur starting from the vortex lock-in speed, and the resulting self-excited vibrations exhibit different phenomenological regimes, depending on their specific characteristics. These regimes are known as the vortex-induced vibration (VIV), galloping, and the combined VIV–galloping. VIV is a resonant phenomenon that arises from the interaction between the Bernard–Karman vortex shedding and the natural vibration frequency of a structure.[3] Galloping, on the other hand, is a type of catastrophic vibration that is caused by the combination of asymmetrical cross sections of bluff body and unsteady airflow perturbations.[4] The VIV–galloping regime represents the fusion of these two regimes, which is facilitated by low mass damping ratios. This allows high-amplitude galloping to take place at low VIV initiation speeds and expands the vibratory range from the limit cycle of resonance to virtually infinite.[5] Assessing the various phenomenological regimes can be beneficial for both passive and active control of flow around bluff bodies to reduce vortex shedding and enhance the surrounding flow. Spot monitoring of flow around bluff bodies can measure velocity and pressure to assess the strength of the vortex street. In this study, a

M. Kazemzadeh, I. Mehdipour, M. De Vittorio, F. Pisanello
Istituto Italiano di Tecnologia
Center for Biomolecular Nanotechnologies
Arnesano (LE) 73010, Italy
E-mail: iman.mehdipour@iit.it, iman.mehdipour@polimi.it

M. De Vittorio, F. Pisanello
Dipartimento di Ingegneria dell'Innovazione
Università del Salento
Lecce 73100, Italy

M. De Vittorio, F. Pisanello
RAISE Ecosystem
Genova, Italy

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/aisy.202300457.

DOI: 10.1002/aisy.202300457

Kármán vortex street-driven membrane triboelectric nanogenerator was utilized for this purpose to monitor ultralow-speed wind energy harvesting and flow sensing.[6] The authors employed a flexible membrane transducer to harness the energy from the vortices and convert it into electrical energy.

Measuring and analyzing the vibratory response of a structure interacting with fluid flow is often challenging due to the complex and flexible geometry of the structure, the presence of strong wind-induced vibrations, unsteadiness, large amplitude vibration, surface slanting, and torsional motion. However, the current measurement and data analysis methods are usually limited to a few points, which may not reflect the entire structure accurately. To tackle these challenges, the utilization of deep learning has emerged as a promising solution in the development of data-driven prediction models for fluid flow research.[7–13] These studies primarily focus on the physical exploration of fluid properties. For instance, in another study,[10] the authors proposed using deep neural networks (DNN) to simulate a fluid dynamics system. Such a system can simulate scenarios much faster than conventional methods that directly deal with partial differential equations. In another set of studies,[9,11,13] deep learning models were employed to enhance the accuracy of conventional simulations by improving their spatial resolution.

There are additional applications of DNNs aimed at directly investigating the impact of fluid on interacting structures. Some examples include the use of the deep neural network-genetic algorithm framework for predicting aerodynamic damping in building engineering,[5] a hybrid partitioned deep learning framework for reducing-order modeling and predicting fluid–structure interaction,[14] and data clustering and random forest regression model for improving the accuracy of predicting VIV on slender marine structures such as deep-water risers.[15,16] Other studies have investigated the use of supervised learning techniques,[17] machine learning algorithms,[18] and a physics-guided deep learning framework[19] to predict the VIVs of real long-span bridges. Additionally, there have been studies on enhancing VIV of a circular cylinder through artificial neural networks trained through deep reinforcement learning,[20,21] reducing hydrodynamic signatures and increasing hydrodynamic stealth through windward-suction-leeward-blowing actuators and data-driven deep reinforcement learning[22] and eliminating VIV of a cylinder through active learning and reinforcement learning.[23,24] Furthermore, DNN approaches have been used for estimating unknown parameters in numerical models of VIV,[25] predicting the dynamic responses of converters and generating optimal power[26] and predicting lift and drag forces in VIV.[27] These approaches have been developed for identifying unknown parameters in highly nonlinear fluid–solid interaction systems.[28–30]

Thus, there is a lack of investigation on the extraction of fluid and turbulence properties through the examination of their vibratory effects on surrounding structures. This requires a distributed sensing system capable of harvesting spatially separated information and patterns. Flexible piezoelectric flags can record such distributed forces and pressure distribution, limited to their length. However, the normal piezoelectric flag configuration only has one output terminal, resulting in highly convoluted electric potential output generated from entire flag. Furthermore, the nonlinear behavior of fluid dynamics, combined with the nonlinear mechanical behavior of the soft piezoelectric flag, adds extra complexities to the problem, making investigation of the system for extracting vortex and turbulence information from this complex system needs a powerful tool like the manifold learning technique.

The exploration of turbulence fingerprinting extends its influence across diverse domains, offering invaluable applications and addressing pressing necessities. In aerospace and aeronautics, the study informs the design of aircraft resilient to turbulent conditions, ensuring both safety and fuel efficiency. The automotive industry benefits from improved aerodynamics and fuel efficiency through turbulence-aware vehicle design. Energy and power generation systems, including wind turbines, are optimized for turbulent wind conditions, enhancing their reliability and output. Environmental modeling gains accuracy in predictions and simulations by incorporating turbulence data, contributing to a deeper understanding of natural systems. In chemical and process industries, turbulence insights aid in designing safe and efficient manufacturing processes. The application extends to heating, ventilation, and air conditioning (HVAC) systems, optimizing airflow and temperature control in buildings for improved energy efficiency and indoor air quality. Additionally, turbulence-aware building design ensures structural integrity and safety in diverse environmental conditions. The necessity of this research lies in fostering safety, reliability, efficiency, environmental impact mitigation, innovation, and cross-disciplinary insights across these vital sectors.

In this work, we propose an autoencoder architecture for[31,32] for manifold learning[33–35] and a data-driven knowledge discovery framework, which is suitable for extracting information and relevant data from such a system. We applied this method to convert the piezoelectric flag and bluff body combination into a real-time wind speed sensor. We also use this technique to classify turbulence generated using different rotation angles of a bluff body, finding resilience to wind speed up to 33 m s^{-1}. We believe such a system will have potential applications in research and industrial fields related to the interaction of a fluid with a solid object, where information needs to be captured from the dynamics of the fluid.

2. Results and Discussion

2.1. The Experimental Model

Our model system is depicted in **Figure 1**a. The schematic model of the VIV configuration includes a bluff body and a clamped piezoelectric flag. As shown in the figure, the bluff body works as a vortex or turbulence generator which induces the flag vibration and consequently generates voltage by the polyvinylidene fluoride flexible (PVDF) piezoelectric membrane. The schematic diagram of a PVDF flag is also presented in Figure 1b. The PVDF membrane, as a piezoelectric layer with the neutral axis in the middle of its thickness, is sandwiched by right and left silver-pasted electrodes and also two laminated layers of Myler (plastic) as protection. The two electrodes are connected to two terminal cables by the rivet electrical connector for data collection.

For an analytical investigation of the effect of the bluff bodies on vortex generations, we have performed a time-domain finite-element method computational fluid dynamic (CFD) simulation via the commercial software package COMSOL. The simulation

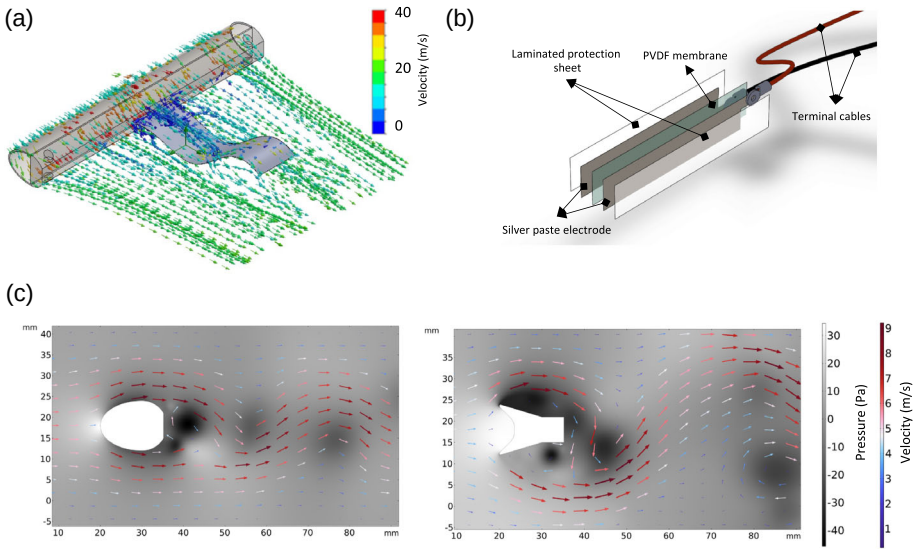

Figure 1. a) The schematic of the VIV configuration. b) Schematic diagram of a PVDF flag. c) CFD simulation results of two bluff body geometries. Heat map indicates the relative air pressure distribution while the arrow lines show the air velocity distribution.

was performed for two different bluff bodies used in this study and their results are displayed in Figure 1c. For this simulation, we considered the incoming air speed starting from 0($m\,s^{-1}$) that gradually increased to 3($m\,s^{-1}$). The calculated wind speed and pressure distribution around the bluff bodies 1 s after reaching 3($m\,s^{-1}$) are depicted as an arrow line and a heat map, respectively. We can clearly see that each bluff body creates different vorticities in different speeds. Theoretically, the shapes, speed, and pressure distribution in these examples should solely depend on the wind speed and the geometry of the bluff bodies.[2] The fluid flow around the bluff body induces vortex shedding, causing the piezoelectric flag to undergo time-dependent movement and generate voltage. This generated charge codifies the local time-dependent deformation of the flag, but its readout can only take place through two electrical connections, resulting in the generation of an electrical potential as a function of the time, hereafter referred to as piezoelectric potential $V(t)$. Still, the flag movements over time should be related to the vorticities, pressure, and wind speed, and this information is intrinsically codified in the $V(t)$ signal. However, as stated before, due to the presence of highly nonlinear physical phenomena stemming from the aerodynamic problem and material properties in the employed components, a mathematical relationship between $V(t)$ and any of the relevant parameters such as bluff body shape, wind speed, or vorticity can hardly be found.

Figure 2a shows the time-domain voltage obtained from the piezoelectric flag attached to the bluff bodies depicted in Figure 1c (bottom right). The voltage was recorded for the entire 370 s of the experiment with a sampling frequency of 48 kHz. To better illustrate the nonlinear behavior of this voltage, a portion of the time-domain signal and its moving root mean square (RMS) (with a window length of 84 ms) are depicted in Figure 2b,c, respectively. As shown in Figure 2b, the voltage exhibits chaotic behavior in short time periods. The moving RMS graph indicates nonlinear trends in the obtained voltage with respect to the wind speed. Specifically, the RMS voltage increases until the wind speed reaches 12 $m\,s^{-1}$, after which it begins to decrease. However, the RMS voltage increases again as the wind speed approaches 20 $m\,s^{-1}$ (refers to Experimental Setup section)

Figure 2d displays the spectrogram of the measured voltage to reveal information about its frequency components. To obtain the spectrogram, the moving Fourier transform of the time-domain signal with a duration of 4096 time-steps (≈84 ms) is calculated and time overlaps are 2048 sampling time-steps (≈42 ms) between each calculation. This produced 8671 spectra samples from our time-domain signal. Then the spectrogram is processed to obtain its baseline and peaks, which are presented in Figure 2e,f, respectively. The baseline of each spectrum was calculated using the asymmetric least square smoothing method.[36]

2.2. Supervised and Unsupervised Manifold Learning using Hybrid Autoencoder

2.2.1. Network Architecture

An autoencoder is employed to investigate the behavior of the spectral data depicted in Figure 2d. The architecture of the autoencoder is illustrated in **Figure 3**a, while Figure 3b,c shows the encoder, decoder, and building blocks of the decoder used in its construction, respectively. In this implementation, all nodes, except for the output layer's nodes, are considered to have a nonlinear activation function known as exponential linear unit. To create a deep enough network capable of learning the complex data topology in the latent projection and avoiding issues like gradient vanishing, we adopted the ResNet[37] architecture for the decoder and wind speed estimator (WS) networks.

It is worth mentioning that we included an additional feed-forward multilayer perceptron network, running parallel to the decoder, to match the obtained latent representation of the data

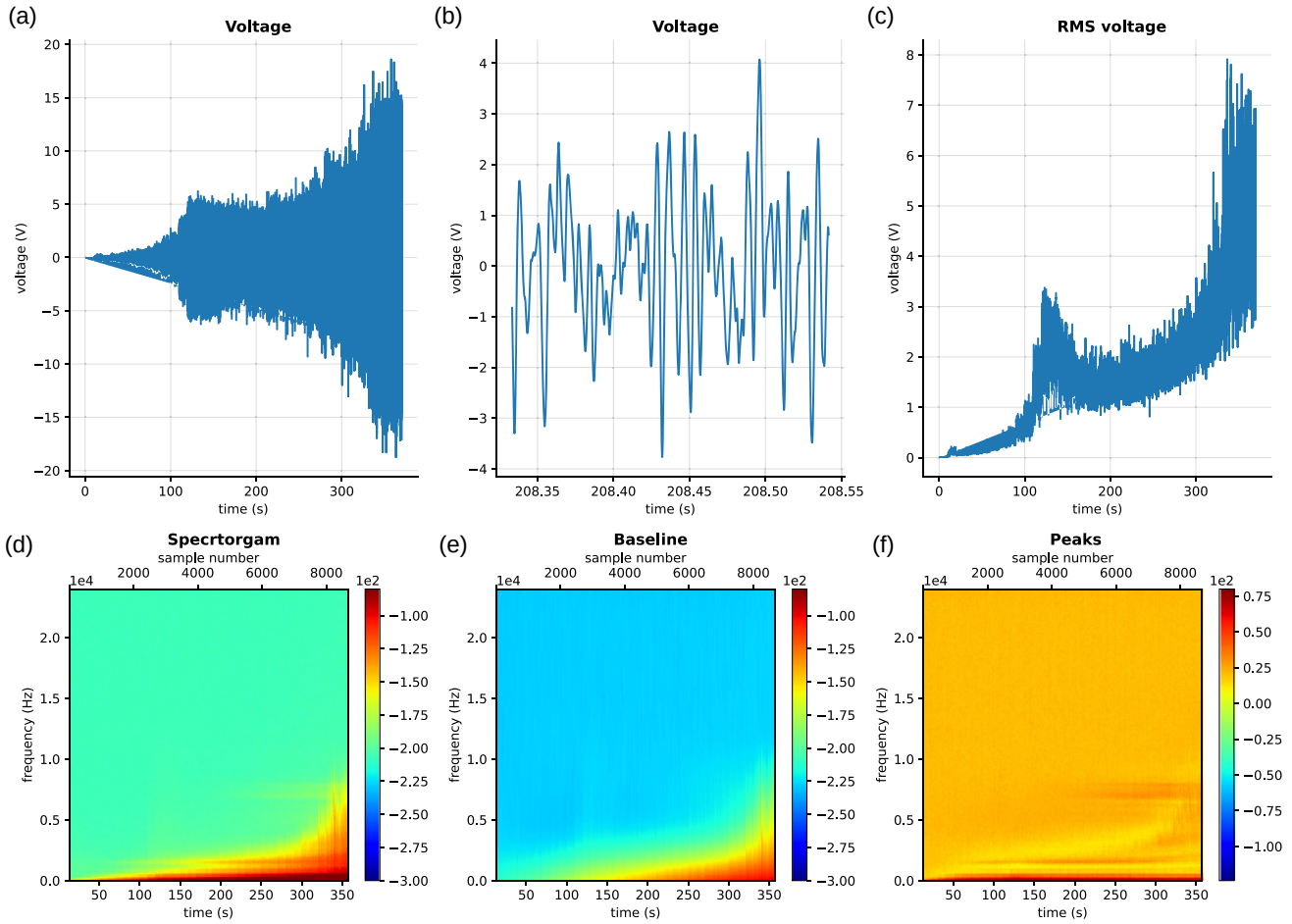

Figure 2. a) The time-domain voltage obtained from the piezoelectric flag when the wind tunnel is commanded with the speed illustrated in Figure 7 (see Section 4). b) A portion of the time-domain voltage that shows its chaotic behaviour. c) Moving RMS of (a). d) The spectrogram of the time-domain voltage in (a). e) Baseline and f) peaks distribution of (e).

from the encoder to the data's label (wind speed in this case). Essentially, the WS penalizes the encoder to find a latent representation that not only ensures good data reconstruction through the decoder but also clearly reflects the wind speed. In some sense, this network is similar to a variational autoencoder, but instead of using the Kullback–Leibler divergence loss to ensure a Gaussian distribution of the data in the latent projection, our method involves penalizing the encoder using the WS.

Although the encoder/decoder network primarily involves unsupervised learning, the incorporation of the wind estimator network introduces a supervised element to the overall system. To achieve purely unsupervised learning outcomes, we assign weights to the losses at the output of both the decoder and the wind estimator. By setting the weight for the wind estimator network to zero, the labels for wind speed cease to influence the encoder network. Consequently, the entire network attains a state of unsupervised learning.

2.2.2. Unsupervised Learning

Learning curves and unsupervised projection of the training and validation spectra, when the latent space only has one dimension, are depicted in **Figure** 4a1–a3, respectively. The color of the projected spectra in Figure 4a2,a3 shows its associated sample number in the original dataset (Figure 2d). So the first spectrum at the beginning of the measurement is numbered 1 and the spectrum at the end of the measurement is numbered 8671. As the wind tunnel is commanded with a staircase function for its operational wind speed, the color has a direct relation to the speed of the wind (both get higher during the measurement). So the points with blue color are associated with lower wind speed and the points with red color are associated with high wind speed. To select the validation set, we randomly separated 20% of the total spectra (8671 spectra) in the dataset. As such we ended up having 6937 spectra in the training set and 1734 in the validation set. We can clearly see that the structure of the obtained manifold of the projection of the training (Figure 4a2) and validation set (Figure 4a3) matches well, indicating the good performance of the autoencoder. Another interesting outcome of this projection is the obtained correlation between the projected value of each spectrum with its associated wind speed; despite this, the unsupervised learning and the network did not have access to any information regarding the wind speed. Additionally, the step-like structure is another interesting

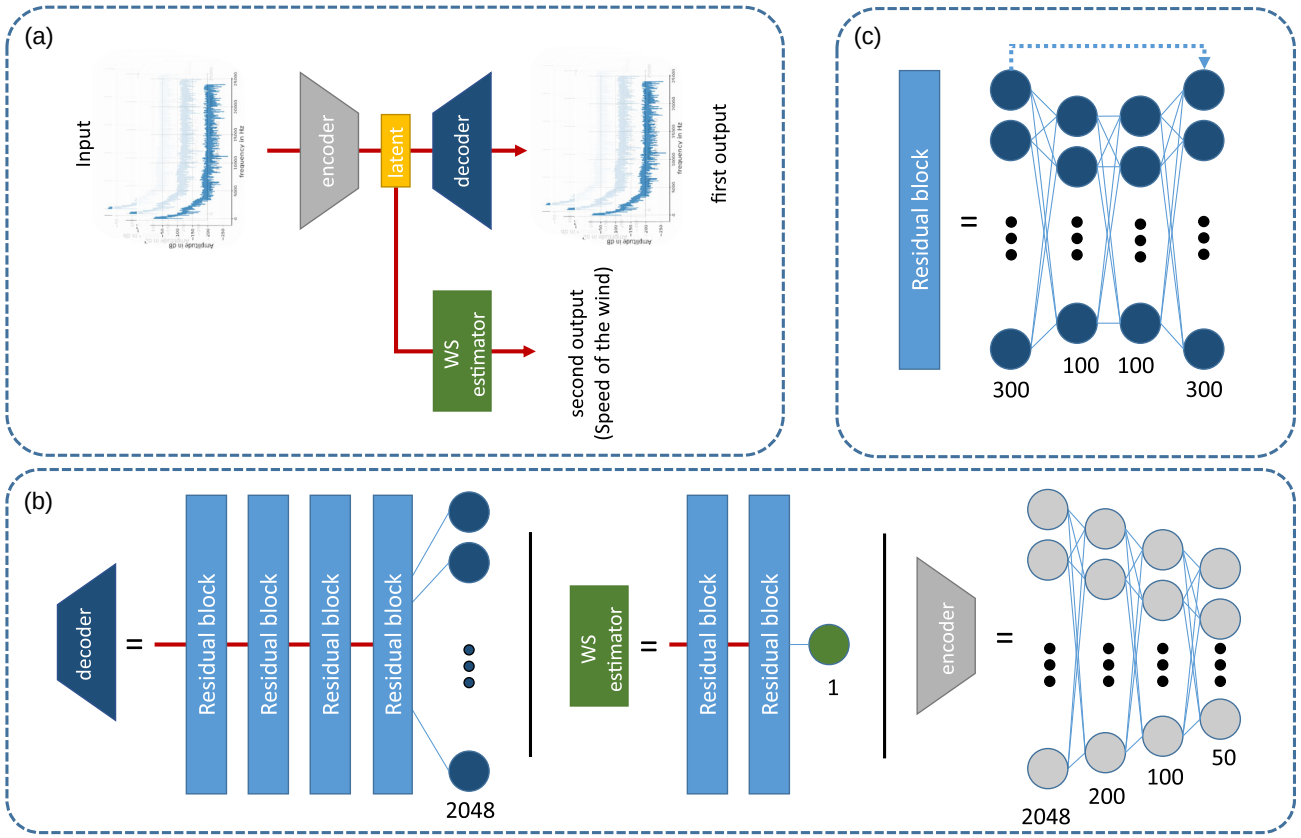

Figure 3. a) The proposed deep learning architecture contains three major parts as encoder, decoder, and WS. b) Architectures of the encoder, decoder, and WS estimator. c) The architecture of the residual block used in the decoder and wind estimator architecture.

feature that can be observed in the spectra projection in Figure 4a2. This step-like structure probably comes from the commended wind speed of the wind tunnel that is a staircase function (Figure 7b) and is more prominent in the spectra associated with lower wind speed (depicted with blue color).

Similar results for 2D and 3D latent space projections are also depicted in Figure 4b1–b3,c1–c3, respectively. Looking at the learning curves depicted in Figure 4, we can see that adding more dimensions into the latent space of this network did not help the network reduce the reconstruction loss even after 100 epochs of training. However, the network was able to achieve a lower loss within a fewer number of epochs of training as we considered a higher number of dimensions for the latent space. We can also see that the projection of the spectra always ended up forming a 1D manifold regardless of the number of dimensions selected for the latent space (Figure 4a2–a3,b2–b3, c2–c3). Interestingly, these extracted 1D manifolds in the latent space are always found to be correlated to the measured wind speed in the wind tunnel.

2.2.3. Supervised Learning

The wind estimator network is incorporated into the decoder network simultaneously and supervised training is conducted. For unsupervised learning, we set the loss weight of the wind estimator network to zero. However, in this case, we specifically chose to set this weight to one-tenth of the decoder's loss. This selection of loss ensures that the latent representation of the data is not only useful for estimating the wind speed but also enables data reconstruction through the decoder. By doing so, we avoid wind speed estimation based on random features in the data that may be caused by noises and disturbances, which improve the reproducibility of the results.

The learning curves, as well as the total training and validation loss for both the network and the WS, are depicted in **Figure 5**a1, a2, respectively. This training was performed using the spectral data shown in Figure 2d. It's important to note that the total loss formula now includes the loss of the decoder (reconstruction loss) plus 0.1 times the loss of the wind estimator. Consequently, the learning curve demonstrates that the network has successfully learned both tasks simultaneously.

The latent representations of both training and validation datasets are illustrated in Figure 5a3,a4, respectively. These graphs show a great improvement in finding the manifold in Figure 4a2,a3 obtained from unsupervised learning. We can observe the staircase-like representation even at the very high sample number (i.e., higher speed of the wind) that only happens at low wind speed in Figure 4a2,a3.

For this problem, we have taken into account two test sets. The first comprises random step functions with varying commanded wind speeds, which lead to significant transitional behavior and

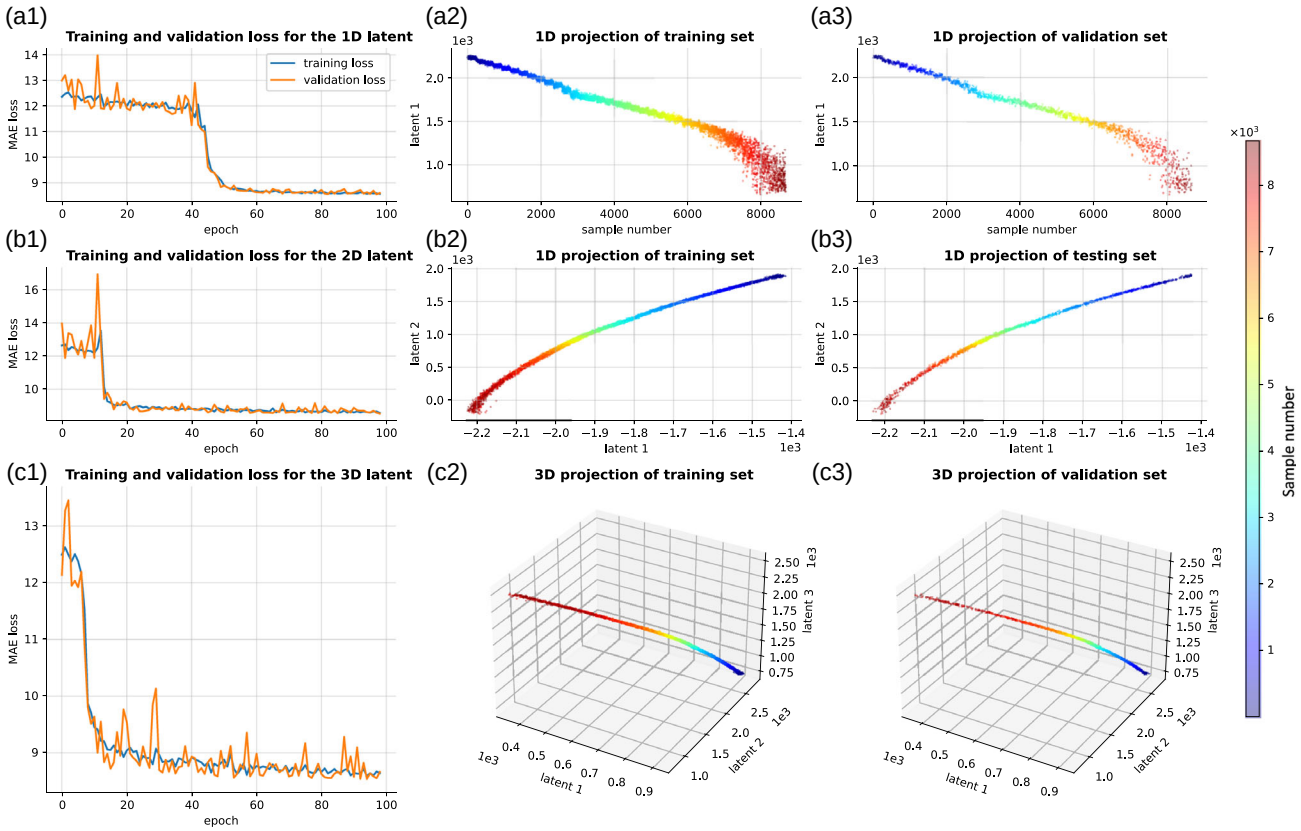

Figure 4. a1,b1,c1) Learning curve when the latent space contains 1D, 2D, 3D, respectively. a2,b2,c2) The projection of the training set in the 1D, 2D, 3D latent space after the training of their corresponding deep learning models. a3,b3,c3) The projection of the validation set in the 1D, 2D, 3D latent space. Color bar shows the sample number illustrated in Figure 2d–f.

sharp wind speed changes. The voltage obtained from the piezoelectric flag and its 1D latent projection using the trained network is depicted in Figure 5b1. The estimated wind speed using our method and the one measured with the pitot tube mechanism are also presented in Figure 5b2. These results suggest that the wind speed calculated using our deep learning method closely follows the measured wind speed while providing real-time behavior as compared to the pitot tube mechanism. This is due to the fact that the pitot tube mechanism can only measure wind speed at 500 ms intervals, while our deep learning model provides measurements at 85 ms intervals.

The distribution of the error between the wind speed measurements obtained from the pitot tube and the deep learning method is illustrated in Figure 5b3. This graph shows that the error distribution is Gaussian. By applying a moving average filter to the deep learning output of the wind speed to match the pitot tube's sampling interval, we can reduce this error. This has been done and the results are plotted in Figure 5b4, where we can see that the error margin has significantly decreased and there is a greater agreement between the pitot tube measurement and deep learning measurement. The same error distribution calculation for Figure 5b4 is depicted in Figure 5b6, showing that the variance of the error distribution is much smaller compared to that in Figure 5b3, as the distribution has been contracted along the error axis.

The commanded wind speed for the second test set simulates the exhaust flow speed of a personal vehicle moving in an urban area. Unlike the previous test set, this set of commanded wind speeds does not exceed 16 $\mathrm{m\,s^{-1}}$ and has smoother changes, resulting in shorter transitional behavior. The voltage versus time obtained from the piezoelectric flag and its latent projection for this problem is depicted in Figure 5c1. The calculated wind speed using the deep learning model and the pitot tube mechanism, as well as its corresponding error distribution, are shown in Figure 5c2,c3, respectively. Interestingly, the variance of the error in this problem appears to be smaller than in the previous test set, due to the more linear and less chaotic behavior of the bluff body and piezoelectric flag at lower wind speeds. A moving average filter has also been applied to match the intervals of the deep learning model to the pitot tube mechanism, as illustrated in Figure 5c4. The associated error distribution is shown in Figure 5c5. It is clear from these results that the error is much smaller when the whole system operates at lower wind speeds.

The proposed system can also be used to extract a manifold by means of the unique characteristics of the turbulence, to identify wind speed and bluff body orientation based on the signal detected by the piezoelectric sensor. Each air flow turbulence has indeed a distinct vortex pattern, spatial distribution of wind speed and direction, and pressure patterns, which contribute to

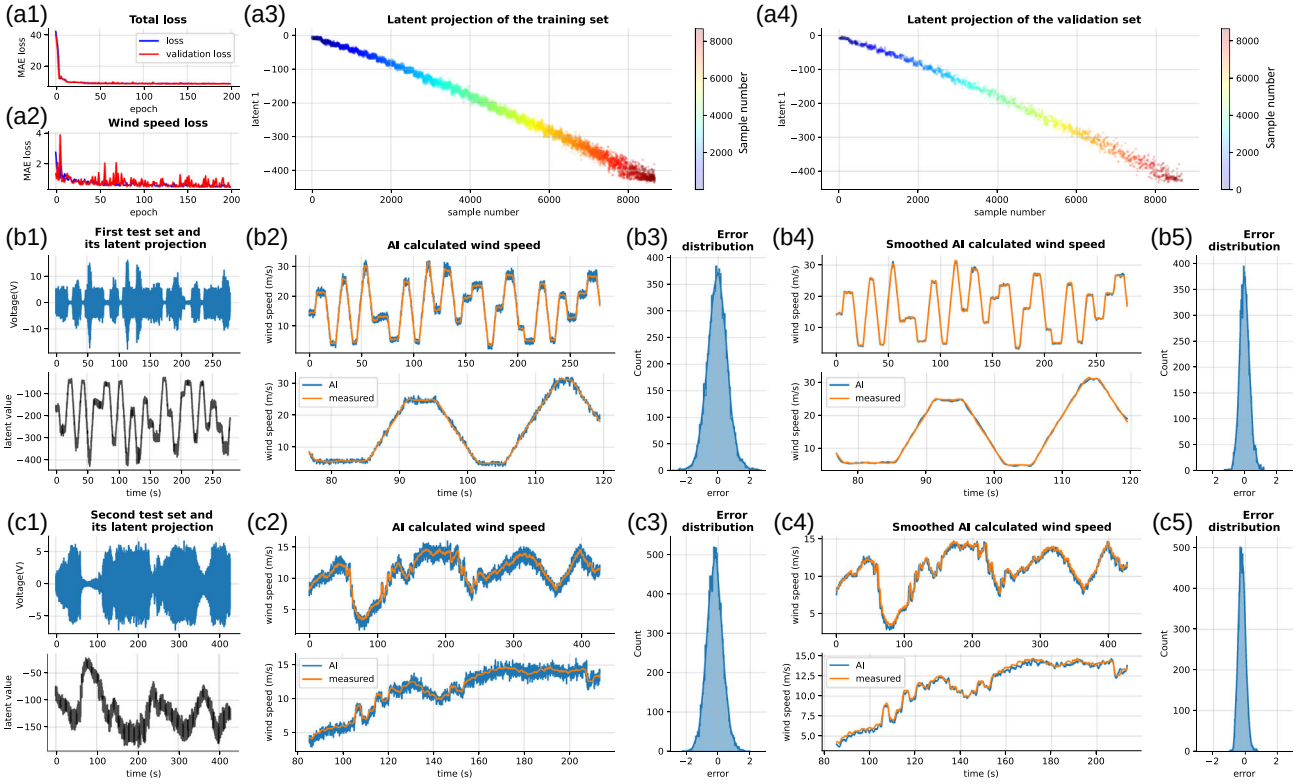

Figure 5. a1,a2) The learning curve associated with the total loss (sum of the decoder and wind estimator output) and wind estimator's loss, respectively. a3,a4) 1D projection of the training and validation set after the training of the deep learning model, respectively. b1) Recorded voltage from the piezoelectric flag of the first test set and its latent projection using the proposed network. b2) Calculated wind speed from the proposed deep learning model. b3) The error distribution associated with the result in (b2). b4) Calculated wind speed using the proposed deep learning model when it is smoothed with moving average to match the output intervals to the one of the pitot tubes. b5) The error distribution associated with the results in (b4). c1) Recorded voltage from the piezoelectric flag of the second test set and its latent projection using the proposed network. c2) Calculated wind speed from the proposed deep learning model. c3) The error distribution associated with the result in (c2). c4) Calculated wind speed using the proposed deep learning model when it is smoothed with moving average to match the output intervals to the one of the pitot tubes. c5) The error distribution associated with the results in (c4).

the sensing paradigm and data analysis described earlier. Indeed, these patterns can be captured nonlinearly by the piezoelectric flag and used to differentiate between different types of turbulence. To show that, we selected the specific bluff body geometry shown in **Figure 6**a1 and its tilted version in Figure 6b1.

Figure 6a1,b1 shows the CFD simulation results, showcasing significant differences in the vortex shedding associated with the two bluff bodies. It's crucial to note that, when considering the same input wind speed and simulation time interval, distinct velocity and pressure distribution patterns emerge. These differences are also reflected in the generated voltage through the piezoelectric flags, as demonstrated in Figure 6a2,b2 for the geometries shown in Figure 6a1,b1, respectively.

To focus solely on the segment characterized by the highest turbulence, we deliberately excluded readings that occurred when the vortexes and turbulence did not predominantly influence the fluid dynamics (i.e., removing data corresponding to slow wind speed and laminar flow). Specifically, we concentrated on the voltage measurements from the last 300 s of the recording, during which the commanded speed aligned with the configuration depicted in Figure 1b.

To apply this data to our deep learning model, we first calculated the spectrogram of the time-domain signal according to the parameters used in the previous section. The spectrograms for the time-domain signal of Figure 6a2,b2 are shown in Figure 6a3, b3, respectively. It is important to note that each spectrogram now contains 7030 spectra since we have 300 s of the time-domain signal. Additionally, we modified the wind estimator network, which previously worked as a regression unit that generates a single real number at its output, to generate classes associated with each bluff body. To do this, we replaced the last layer of the wind estimator with a layer consisting of two nodes and used the softmax activation for it. This network can perform classification based on the assigned labels to each of the datasets depicted in Figure 6a3,b3. Since we are dealing with two independent parameters in this problem, we selected the dimension of the latent space to be equal to 2. The wind speed, which changes continuously, is the first independent parameter, and it leads to the presence of a continuous 1D manifold. The second independent parameter is the bluff body itself. We expect the network to extract two continuous 1D manifolds in its latent space when it is trained on these dataset.

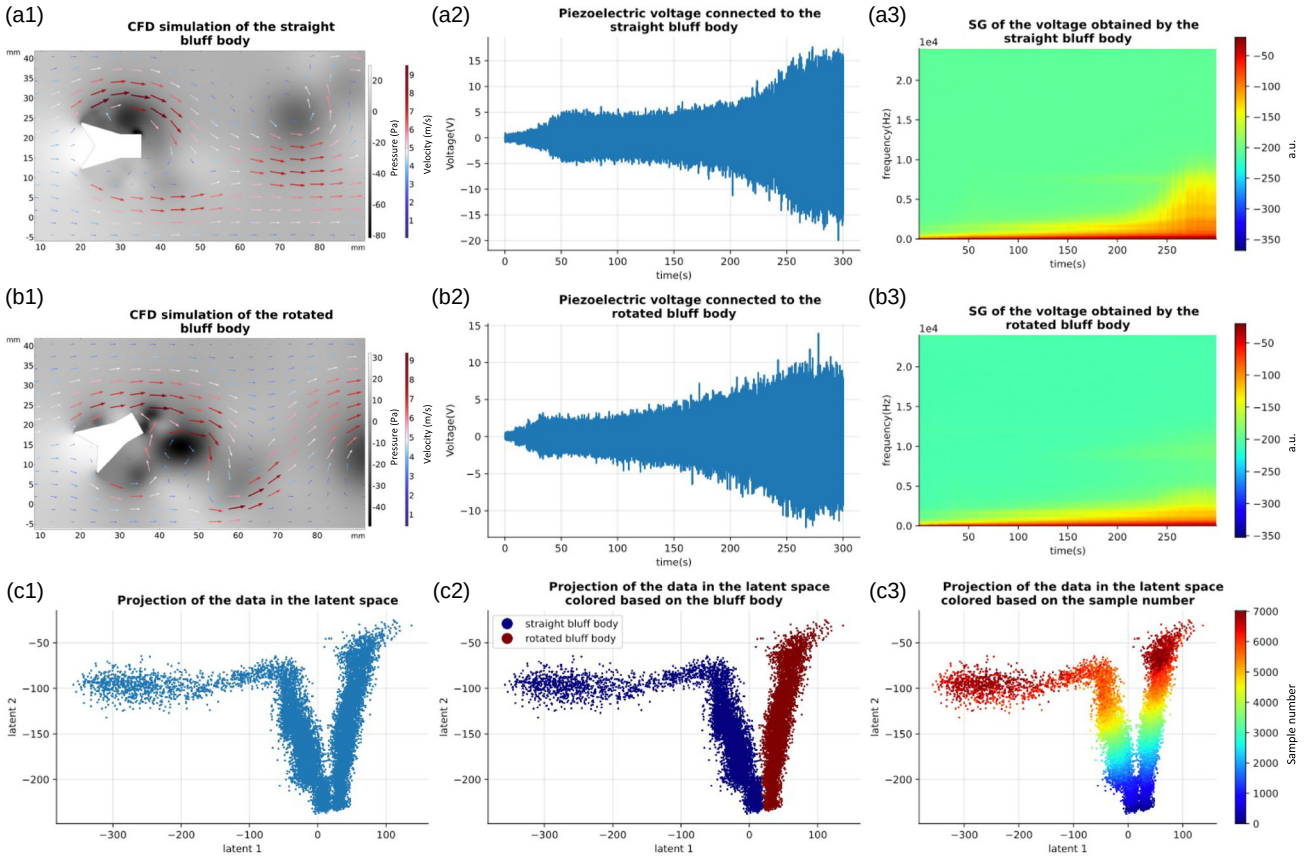

Figure 6. a1,b1) CFD simulation of the two different bluff bodies. a2,b2) The associated voltage obtained from bluff bodies in (a1) and (b1), respectively. a3,b3) The spectrogram of the time-domain voltage in (a2) and (b2), respectively. c1) Projection of the spectra in (a3) and (b3) using the proposed network. c2) The projected spectra when each projected point is colored based on the bluff body origin. c3) The projected spectra when each projected point is colored based on the sample number.

The projection of all spectra of this dataset is shown in Figure 6c1. These points are colored based on their bluff body and are also shown in Figure 6c2. We can clearly see that the spectra of one bluff body are fully separated from the spectra of the other. We can also color code each projected spectrum based on the sample number or time from the beginning of the experiment. This result is depicted in Figure 6c3. Interestingly, the extracted manifold of these spectrograms is highly correlated with the sample number or in other words, highly correlated with the wind speed of the wind tunnel, supporting the hypotesis that the approach is resilient to the windspeed.

3. Conclusion

In this work, we introduced deep learning architecture to classify turbulences detected by a flexible piezoelectric MEMS. To achieve this, we developed a deep learning method based on autoencoder architecture to extract and project a continuous manifold associated with turbulence. This method allows extracting the wind speed though the piezoelectric flag is placed in a highly turbulent region and captures the differences in turbulence generated by different bluff bodies with high resilience to wind speed up to maximum of our measurment which is 33 ($m\,s^{-1}$). Its ability to operate in the wide windspeed range makes the technique highly applicable in a broad range of fields that require an understanding of turbulence engineering, including aerospace and aeronautics, automotive industry, energy and power generation, environmental modeling, chemical and process industries, HVAC, and building design and environmental fluid dynamics.

4. Experimental Section

Experimental Setup: The test chamber of the wind tunnel was carefully configured to ensure laminar flow and repeatability for any given configuration, achieved through the implementation of a specific wind flow function.[2] The experimental setup involves utilizing a variety of proposed bluff body shapes in combination with various flag configurations.

For conducting the experimental tests, the wind tunnel lab uses a subsonic open-loop wind tunnel model named AF100. Air was propelled by a centrifugal fan, ensuring a fairly uniform wind speed within the wind tunnel's cross-section. The dimensions of the test chamber were 35 cm × 35 cm, and its smooth walls helped minimize the boundary layer thickness. To measure the wind speed generated in the tunnel, two pitot tubes were employed, which took into account the test environment

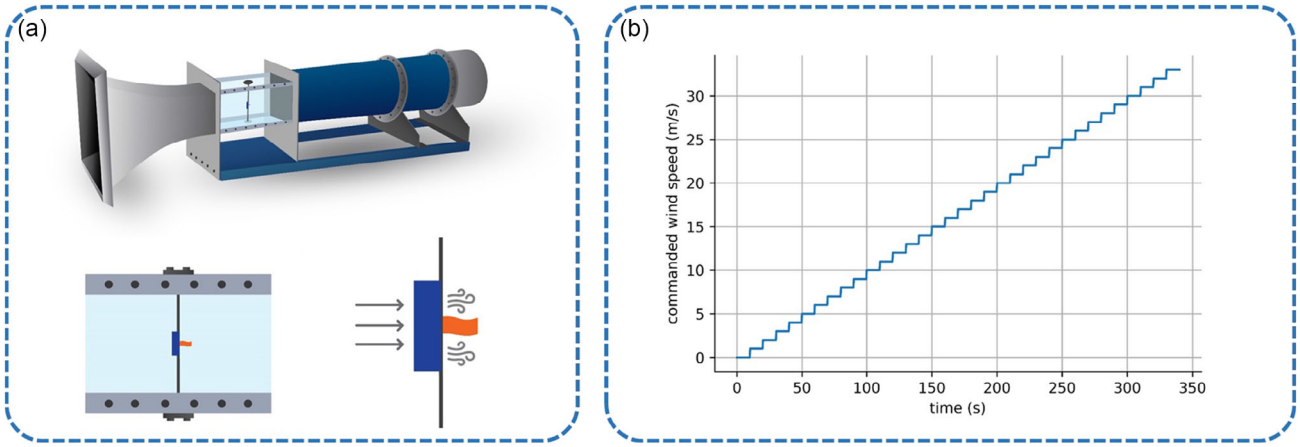

Figure 7. a) The schematic of the experimental setup including the wind tunnel and configuration of the bluff body and the piezoelectric flag within the air tunnel chamber. b) Commanded wind speed to the wind tunnel.

temperature and the zero-reference pressure at the test's start. An Arduino board and stepper motor automated the manual speed controller, which was precisely calibrated.

Within the wind tunnel's test chamber, a piezoelectric flag was clamped onto a vertically fixed aluminum column, as illustrated in **Figure 7a**. A wind straightener created a laminar flow in the wind tunnel, where a bluff body generated vortexes. These vortexes then vibrated the piezoelectric flag and generated electrical potential in the flag's terminals. The commanded wind speed to the wind tunnel is also presented in Figure 7b. We selected the staircase function with a step width equal to 10(s) and a step height of the 1($m\,s^{-1}$). The relatively long step width was to let the rotary motor of the wind tunnel reach its steady state and avoid lagging behind the commanded wind speed in the tunnel. The piezoelectric flag (DT4-052 K L^{-1} w/rivets) schematic diagram, dimensions, and material properties are given in Figure 1b and **Table 1**, respectively. Moreover, different bluff bodies were affixed to the aluminum column using two pins.[2] These bluff body shapes were designed cylindrically, featuring varying cross sections, while maintaining a uniform length of 10 cm.

Table 1. Parameters of the flag.

Parameters	Value
Flag total length [mm]	72
Flag active length [mm]	62
Width of flag [mm]	16
Active width of flag [mm]	12
PVDF membrane thickness [μm]	28
Flag total thickness [μm]	157
Young's modulus [10^9 N m^{-2}]	2–4
Poisson's ratio	0.46
Density of flag [kg m^{-3}]	1.78
Relative permittivity [$\varepsilon/\varepsilon_0$]	12
d_{31} constant [10^{-12} C N^{-1}]	23
g_{31} constant [10^{-3} Vm N^{-1}]	216
k_{31} constant [% at 1 kHz]	12
Capacitance [pF cm^{-2} at 1 kHz]	380
Output voltage [V]	0.01–100
Impedance [MΩ]	1.0

Deep Learning Training: All the deep learning architectures were implemented using the TensorFlow[38] library in Python, and the training was performed on a single NVIDIA graphics processing unit, specifically the 3090ti model with 24 gigabytes of graphical memory. Each network architecture was trained in less than half an hour, with fewer than 200 training epochs. Additionally, each spectrum preprocessing step takes only a few milliseconds when utilizing any of the trained networks presented.

Acknowledgements

M.K. and I.M. contributed equally to this work. M.D.V. and F.P. supervised and are co-last authors. M.D.V. and F.P. acknowledge funding from the European Union's Horizon 2020 Research and Innovation Program under grant agreement no. 828972. M.D.V. and F.P. acknowledge that this project has received funding from the European Union's Horizon 2020 Research and Innovation Program under grant agreement no. 101016787. M.D.V. and F.P. acknowledge funding from project "RAISE (Robotics and AI for Socio-economic Empowerment)" code ECS00000035 funded by the European Union - NextGenerationEU PNRR MUR - M4C2 - Investimento 1.5 - Avviso "Ecosistemi dell'Innovazione" CUP J33C22001220001. This research was supported by fishRISE Project funded by Programma Operativo Nazionale "Ricerca e Innovazione" 2014–2020 (PON "R&I" 2014–2020) - "Remote, Intelligent & Sustainable aquaculturE system for Fish (fishRISE)" n. ARS01_01053."

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

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

Keywords

autoencoders, deep learning, fluid dynamics, manifold learning, piezoelectric energy harvesting, turbulence fingerprinting, vortex-induced vibrations

Received: July 31, 2023
Revised: October 24, 2023
Published online: November 30, 2023

[1] M. Mariello, F. Guido, V. M. Mastronardi, F. Madaro, I. Mehdipour, M. T. Todaro, F. Rizzi, M. De Vittorio, *Nano Tools and Devices for Enhanced Renewable Energy* (Ed: S. Devasahayam), Elsevier **2021**, pp. 291–374, https://doi.org/10.1016/B978-0-12-821709-2.00023-2.
[2] I. Mehdipour, F. Madaro, F. Rizzi, M. De Vittorio, *Energy Convers. Manage.: X* **2022**, *13*, 100174.
[3] Z. Chen, K. Tse, K. Kwok, B. Kim, A. Kareem, *Eng. Struct.* **2020**, *202*, 109855.
[4] Z. Chen, H. Huang, K. Tse, Y. Xu, C. Y. Li, *J. Wind Eng. Ind. Aerodyn.* **2020**, *205*, 104325.
[5] Z. Chen, L. Zhang, K. Li, X. Xue, X. Zhang, B. Kim, C. Y. Li, *J. Build. Eng.* **2023**, *63*, 105374.
[6] W. Li, L. Lu, X. Fu, C. Zhang, K. Loos, Y. Pei, *ACS Appl. Mater. Interfaces* **2022**, *14*, 51018.
[7] J. M. Alam, *Fluids* **2023**, *8*, 224.
[8] M. Majchrzak, K. Marciniak-Lukasiak, P. Lukasiak, *Energies* **2023**, *16*, 1755.
[9] K. Fukami, K. Fukagata, K. Taira, *Theor. Comput. Fluid Dyn.* **2023**, *37*, 421.
[10] D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, S. Hoyer, *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2101784118.
[11] B. Liu, J. Tang, H. Huang, X.-Y. Lu, *Phys. Fluids* **2020**, *32*, 2.
[12] J. Zhu, H. Zhang, Z. Hu, X. Lu, Q. Zhan, Y. Cai, C. Zhao, *Appl. Phys. Lett.* **2023**, *123*, 9.
[13] J. Zhang, J. Liu, Z. Huang, *Ocean Eng.* **2023**, *280*, 114902.
[14] R. Gupta, R. Jaiman, *Comput. Fluids* **2022**, *233*, 105239.
[15] S. Riemer-Sørensen, J. Wu, H. Lie, S. Sævik, S.-W. Kim, in *Nordic Artificial Intelligence Research and Development: Third Symp. of the Norwegian AI Society (NAIS 2019)*, Trondheim, Norway, May 27–28, **2019**, Springer, pp. 78–89.
[16] J. Wu, D. Yin, H. Lie, S. Riemer-Sørensen, S. Sævik, M. Triantafyllou, *J. Mar. Sci. Eng.* **2020**, *8*, 127.
[17] J. Lim, S. Kim, H. K. Kim, *J. Wind Eng. Ind. Aerodyn.* **2022**, *221*, 104904.
[18] S. Kim, T. Kim, *Eng. Struct.* **2022**, *266*, 114551.
[19] S. Li, S. Laima, H. Li, *Phys. Fluids* **2021**, *33*, 037113.
[20] Y. F. Mei, C. Zheng, N. Aubry, M. G. Li, W. T. Wu, X. Liu, *Phys. Fluids* **2021**, *33*, 103604.
[21] C. Zheng, F. Xie, T. Ji, X. Zhang, Y. Lu, H. Zhou, Y. Zheng, *Phys. Fluids* **2022**, *34*, 113603.
[22] F. Ren, C. Wang, H. Tang, *Phys. Fluids* **2021**, *33*, 093602.
[23] C. Zheng, T. Ji, F. Xie, X. Zhang, H. Zheng, Y. Zheng, *Phys. Fluids* **2021**, *33*, 063607.
[24] Y. Ran, Z. Deng, H. Yu, W. Chen, D. Gao, *J. Vis.* **2023**, *26*, 1.
[25] T. Misaka, *Struct. Multidiscip. Optim.* **2020**, *62*, 805.
[26] G. Y. Kim, C. Lim, E. S. Kim, S. C. Shin, *Appl. Sci.* **2021**, *11*, 7163.
[27] M. Raissi, Z. Wang, M. Triantafyllou, G. Karniadakis, *J. Fluid Mechanics* **2019**, *861*, 119.
[28] D. H. Lim, K. S. Kim, *J. Sound Vib.* **2022**, *520*, 116629.
[29] Y. Wu, Z. Cheng, R. McConkey, F.-S. Lien, E. Yee, *Energies* **2022**, *15*, 8719.
[30] C. Cheng, H. Meng, Y. Z. Li, G. T. Zhang, *Ocean Eng.* **2021**, *240*, 109932.
[31] D. P. Kingma, M. Welling, (Preprint) arXiv:1312.6114, Submitted on 20 Dec 2013 (v1), last revised 10 Dec 2022 (v11), **2013**.
[32] Y. Wang, H. Yao, S. Zhao, *Neurocomputing* **2016**, *184*, 232.
[33] L. McInnes, J. Healy, J. Melville, (Preprint) arXiv:1802.03426, Submitted on 9 Feb 2018 (v1), last revised 18 Sep 2020 (v3), **2018**.
[34] L. Van der Maaten, G. Hinton, *J. Mach. Learn. Res.* **2008**, *9*, 11.
[35] A. F. Duque, S. Morin, G. Wolf, K. Moon, in *2020 IEEE Int. Conf. on Big Data (Big Data)*, IEEE, Atlanta, GA, USA **2020**, pp. 5027–5036, https://doi.org/10.1109/BigData50022.2020.9378049.
[36] J. Peng, S. Peng, A. Jiang, J. Wei, C. Li, J. Tan, *Anal. Chim. Acta* **2010**, *683*, 63.
[37] K. He, X. Zhang, S. Ren, J. Sun, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA **2016**, pp. 770–778, https://doi.org/10.1109/CVPR.2016.90.
[38] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, et al., *Tensor Flow: Large-Scale Machine Learning on Heterogeneous Systems* **2015**, https://doi.org/10.1109/CVPR.2016.90 https://www.tensorflow.org/.