# Topological Data Analysis Applied to Wind Turbine Vibration Spectra for Blade Icing Detection

Álvaro Martín Gómez, Thomas Haugaard, Oier Ajenjo de Torres, Torben Knudsen, Yossi Bokor Bleile, Rafał Wiśniewski

***Abstract*—Ice buildup on wind turbine blades is a significant issue, leading to operational risks and reduced efficiency. Conventional ice detection methods, such as visual inspection, power curve analysis or specialised sensors, are often slow, inefficient, or costly. This paper proposes an approach using 0-dimensional persistence homology from topological data analysis (TDA) applied to tower and blade vibration spectra. This method extracts key features representing the lifespan of the sub-level sets of the spectra, allowing the formulation of a clearer supervised learning problem. The resulting persistence diagrams are embedded into persistence images and persistence rank functions. Persistence images are employed alongside convolutional neural networks (CNN) to distinguish asymmetrical ice distribution on one or two blades as well as symmetrical ice distribution across three blades from normal conditions. For the symmetrical ice distribution scenario, persistence rank functions with functional principal component analysis (FPCA) and support vector machines (SVM) offer a simpler classification. This approach not only improves ice detection accuracy but also reduces equipment costs and maintenance, promising enhanced wind turbine blade monitoring and maintenance efficiency.**

***Note to Practitioners*—This study is motivated by the need for an accurate ice detection strategy in wind turbine blades that ensures precise and reliable detection. It utilises standard data from wind turbine structural vibrations to avoid additional expenses associated with specialised sensors. Unlike traditional faults, the gradual accumulation of ice on wind turbine blades permits the use of more intricate data transformations such as topological data analysis (TDA), which would otherwise be too computationally intensive for anomalies requiring detection within seconds. Additionally, this approach enables a more refined detection algorithm, crucial for identifying cases in which the ice distribution is symmetric across the wind turbine rotor, which does not produce imbalance loads in the structure and therefore has a subtler effect.**

***Index Terms*—Topological data analysis, anomaly detection, wind energy, time-series analysis, machine learning.**

## I. Introduction

This work was funded by Innovation Fund Denmark.

Álvaro M. Gómez, Torben Knudsen and Rafał Wiśniewski are with the Department of Electronic Systems, Aalborg University, Fredrik Bajers Vej 7, 9220 Aalborg, Denmark (e-mail: almago@es.aau.dk; tk@es.aau.dk; raf@es.aau.dk).

Thomas Haugaard is with Emerson, Håndværkervej 1, Rødkærsbro, Denmark (e-mail: Thomas.Haugaard@emerson.com).

Oier Ajenjo de Torres did his M.Sc. in Control and Automation at Aalborg University, Aalborg, Denmark (e-mail: ajentoier@gmail.com).

Yossi Bokor Bleile is with the Department of Mathematical Sciences, Aalborg University, Skjernvej 4A, 9220 Aalborg, Denmark (e-mail: yossib@math.aau.dk).

ATMOSPHERIC icing on wind turbines encompasses two main types [1]: in-cloud icing (rime ice or glaze) and precipitation icing (freezing rain or drizzle, wet snow), each with distinct characteristics and formation conditions. Rime ice, occurring at temperatures between 0°C and -20°C, consists of supercooled water droplets from clouds or fog that freeze upon impact. It forms asymmetrically as soft or hard rime, and it is distinguished by its irregular crystalline structure and adherence to surfaces. Glaze, induced by freezing rain or in-cloud icing, creates a smooth, transparent ice layer with strong adhesion, typically at temperatures between 0°C and -6°C. Wet snow accretion, formed by sticky partly melted snow crystals, occurs between 0°C and +3°C, with the potential to freeze upon temperature decrease. These different forms of atmospheric icing present varied densities and adhesion properties, impacting wind turbine development and posing challenges for mitigation and removal.

Icing on wind turbine blades can significantly reduce their efficiency by altering their aerodynamic profile. The added weight of the ice can also cause imbalance issues, leading to increased stress on the turbine components and potentially affecting its structural integrity. Additionally, ice shedding from the blades can pose a safety hazard to nearby personnel and property. Visual inspections or remote monitoring systems equipped with cameras are typically used to directly observe the condition of the turbine blades for signs of ice accumulation [2]. These methods may not be practical for continuous monitoring, especially in remote locations or during adverse weather conditions. Changes in power output or turbine performance metrics, such as rotor speed or pitch angle deviations, can also indicate the presence of ice on the blades affecting aerodynamic efficiency, provided that the variables being monitored are not influenced by factors other than icing [3]–[5]. Issues associated with data management, such as the need for large data storage, vulnerability to cyberattacks, and unwillingness of data owners to share their data, are tackled in [6] by training of deep learning models in a distributed manner via a federated learning framework. Lastly, specialised icing sensors may be installed on the blades to directly detect the presence of ice [7]. These sensors can utilise various technologies such as capacitance, infrared, or ultrasonic methods to detect ice buildup. Their main issue is that they require additional equipment and installation costs.

This study aims to achieve accurate and cost-effective icing diagnosis by exploiting the available SCADA data. SCADA

data provides real-time monitoring of various variables related to wind turbine operation and structural dynamics. These data offer valuable insights into the condition of the wind turbine, without incurring extra expenses. A series of mathematical transformations are subsequently applied to tower deflection and blade root bending moment signals to enhance the clarity of the diagnosis. First, the proposed method computes the fast Fourier transform (FFT) of the mentioned variables in order to identify abnormalities in the distribution and amplitude of the harmonics. Tower deflection dynamics will allow us to see abnormal vibrations due to the asymmetric ice distribution in the rotor, while blade root bending moments will show more clearly the slight increase in the amplitude of individual blade vibrations even if the ice is evenly distributed in the three blades. The novelty of this paper resides in the additional feature extraction method applied to the spectra, based on 0-dimensional (sub-level set) persistent homology, a technique from topological data analysis (TDA). The objective is to automatically find out a reduced set of features by encoding the topological information of the sub-level sets of the spectra, so that the decision-making process is performed in that new feature space instead of directly using the spectra. This information can be used to compute different topological summaries which are amenable to be used by machine learning algorithms. Two supervised learning frameworks will be compared based on two different topological summaries: image classification of persistence images using convolutional neural networks (CNN) and functional principal component analysis (FPCA) along with support vector machines (SVM) applied to persistence rank functions. To the knowledge of the authors, this is the first time sub-level set homology is used as feature extractor in FFT spectra for time-series anomaly detection. No prior applications of this technique have been found within industrial contexts, particularly in the field of energy production.

While modern wind turbines have blade root sensors installed, tower deflection is typically estimated with a Kalman filter using accelerometer measurements (among others) together with a standard model comprising the main wind turbine dynamics. The Normalised Unscented Kalman filter (NUKF) derived in [8] is used for state estimation to avoid issues regarding ill-conditioned covariances due to the presence of both large and small states (e.g. generator torque compared to pitch angle).

A $15\,\mathrm{MW}$ wind turbine simulation model in the software GH Bladed was provided by Emerson to generate data in both normal and icing conditions for the training of the machine learning algorithms as well as for model validation. GH Bladed facilitates the simulation of icing conditions in one, two and three blades separately. Moreover, different wind conditions were used to account for various degrees of turbulence, and wind speeds ranging from $8\,\mathrm{m/s}$ to $24\,\mathrm{m/s}$ were considered. This will guarantee that the achieved accuracy reflects real-world conditions and that the method can be effectively applied in practical scenarios.

## II. Problem Settings

The IEC 61400-1 standard [9] defines three wind classes, denoted as I, II, III, and further subdivided into A, B, and C categories. Each class is defined based on particular wind conditions characterised by $v_{ref}$, representing the maximum $10\,\mathrm{min}$ average wind speed over a 50-year period, and turbulence intensity $t_i$, which measures the typical variation in wind within a $10\,\mathrm{min}$ timeframe. This classification is shown in Table I. Class S corresponding to special wind or other external conditions has not been included.

TABLE I: Basic parameters for wind turbine classes [9]

| **Wind class** | | **I** | **II** | **III** |
|---|---|---|---|---|
| $v_{ref}$ [m/s] | | 50 | 42.5 | 37.5 |
| A | $t_i$ [·] | | 0.16 | |
| B | $t_i$ [·] | | 0.14 | |
| C | $t_i$ [·] | | 0.12 | |

In this case, the distinction between class I, II, and III is not relevant, as only wind speeds within the operational range of the wind turbine are being simulated. Additionally, class A has been omitted from simulations due to the limitations of the wind turbine components' design. Including such turbulence would exceed the turbine's load design limits, requiring component reinforcement, which would change modal response and increase costs. Consequently, simulations for classes IB and IC will be used.

Furthermore, two distinct turbulence models have also been considered: normal turbulence model (NTM), which represents the normal range of turbulence conditions experienced by a wind turbine over time, and extreme turbulence model (ETM), which represents the extreme or severe turbulence conditions that a wind turbine might encounter sporadically. ETM is commonly used to analyse the response of the turbine to these extreme events and to ensure that the turbine can withstand such conditions without failure. Check [9] for a detailed definition of the models for the different wind conditions. As ETM is considered in the simulations, GH Bladed sets a minimum wind speed of $8\,\mathrm{m/s}$ to ensure the presence of extreme turbulence is meaningful. The maximum wind speed value for safe wind turbine operation is set to $24\,\mathrm{m/s}$.

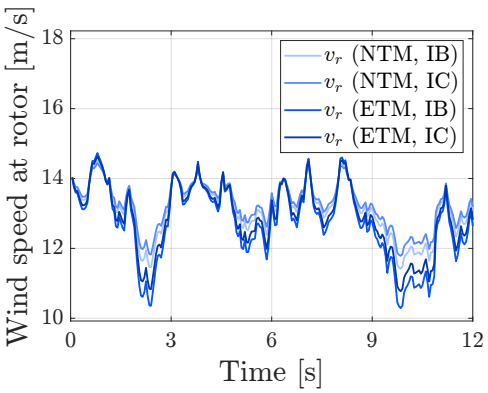


Fig. 1: Comparison between the different wind turbulence models and wind classes simulated in GH Bladed for a simulation with average mean wind speed of $14\,\mathrm{m/s}$.

To simulate ice accretion on the blades, the mass distribution of the blade can be adjusted accordingly. Following the guidelines outlined in [10], which detail calculations for the presence of ice in the rotor, three scenarios will be explored: ice formation on one, two, and three blades. These ice distribution scenarios produce different effects in blade and tower vibrations [11]: mass imbalance resulting from asymmetrical ice mass distribution on the blades considerably increases the fatigue on the wind turbine structure, while symmetrical ice mass distribution has a much more subtle effect, as seen in Fig. 3. The ice mass distribution, denoted as mass per unit length, is assumed to be concentrated at the leading edge. Starting from zero at the rotor axis, it linearly increases to a magnitude represented by $\mu_e$ at half the radius, maintaining a constant value up to the outermost radius. The value of $\mu_e$ is determined by [10]:

$$\mu_e = \rho_e \, k \, c_{min} \, (c_{max} + c_{min}) \tag{1}$$

Here, $\mu_e$ [kg/m] represents the mass distribution at the leading edge of the rotor blade at half the rotor radius. Parameters include ice density ($\rho_e = 700\,\mathrm{kg/m^3}$), and the coefficient $k$, defined as:

$$0.00675 + 0.3\, e^{-0.32\, R/R_1} \tag{2}$$

where $R$ represents the rotor radius, $R_1 = 1\,\mathrm{m}$, $c_{max}$ stands for the maximum chord length, and $c_{min}$ is the chord length at the blade tip, extrapolated linearly from the blade contour as seen in Fig. 2.

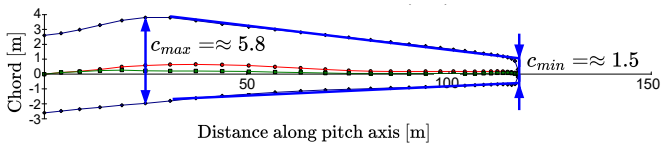


Fig. 2: Maximum and minimum chord length for the GH Bladed 15 MW wind turbine model. The red and green curves represent the centre of mass and shear centre respectively.

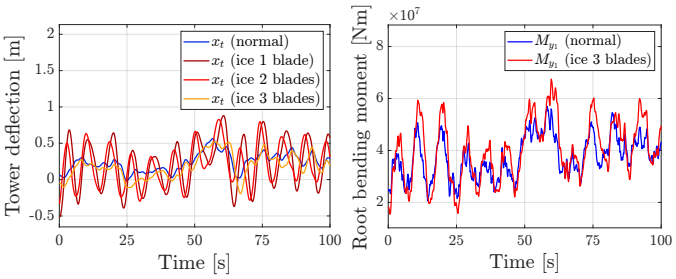


Fig. 3: Normal scenario and the three icing scenarios considered for tower foreaft deflection (left), normal scenario and symmetrical ice mass distribution in 3 blades for blade flapwise root bending moment (right). Both simulations have an average mean wind speed of 14 m/s.

## III. Proposed Strategy

The detection algorithm utilises 10 min windows of data from tower foreaft deflection and blade flapwise root bending moments, obtained from GH Bladed simulations for a 15 MW wind turbine model. The simulation sample time is 0.05 s. The different wind conditions and classes mentioned in Section II are used to account for all possible scenarios and to achieve realistic results. Tower foreaft[1] deflection data helps distinguishing between normal conditions and asymmetrical ice mass distribution scenarios (that is, ice accumulation on one or two blades). Conversely, data from root bending moments enables more precise detection in cases of symmetrical ice mass distribution (that is, ice present on all three blades). While modern wind turbines typically provide direct measurements for root bending moments, tower deflection must be estimated via Kalman filtering. This estimation involves using tower acceleration measurements obtained from an accelerometer in conjunction with a standard aerodynamic and structural model, along with measurements for wind speed, rotor speed, power, and pitch angle. Further details on this estimation process are provided in Section IV.

The subsequent procedure is illustrated in the workflow outlined in Fig. 4. Signals from the two variables under consideration are subjected to frequency domain analysis to achieve a clearer and more stable depiction of the impact of ice on the blades. Subsequently, sub-level set homology from TDA is applied to extract features that capture the most significant harmonics by analysing the lifespan of sub-level sets of the spectra across increasing levels. The outcome of this analysis is then represented through two topological summaries: persistence images and persistence rank functions, which are amenable to machine learning algorithms. The processed data is organised into dictionaries, with fields corresponding to different wind speeds. This procedure is elaborated in detail in Section V, and it has been implemented in Julia using RIPSERER.JL [13] for the generation of cubical complex filtrations and the corresponding persistence diagrams, PERSISTENCEDIAGRAMS.JL [14] for the persistence image embedding, and DISCRETEPERSISTENTHOMOLOGYTRANSFORM.JL [15] for the persistence rank function embedding and subsequent application of FPCA.

The two computed topological summaries are used in two different schemes. Section VI delves into the classification of persistence images using CNNs (implemented with KERAS in Python), whereas Section VII treats each persistence rank function as a score in the principal component space using FPCA. Then, these scores are used to define decision regions distinguishing normal conditions from those affected by ice using SVM. A subsequent discussion compares both approaches.

It is important to remark that tower and blade dynamics will significantly change depending on the mean wind speed being considered. As the range of wind speeds increases, it becomes more challenging for a unique machine learning algorithm to generalise effectively, which decreases accuracy

[1]Note that tower side-to-side deflection is more commonly used for detecting excessive oscillations due to an unbalanced rotor [12]. However, there is not much difference in comparison to foreaft deflection if the rotor remains balanced due to a symmetrical ice mass distribution. The same procedure presented in this paper could have been applied to tower side-to-side spectra, but taking into account that unbalanced oscillations are easily seen in both, it was decided to keep the focus on out-of-plane vibrations.

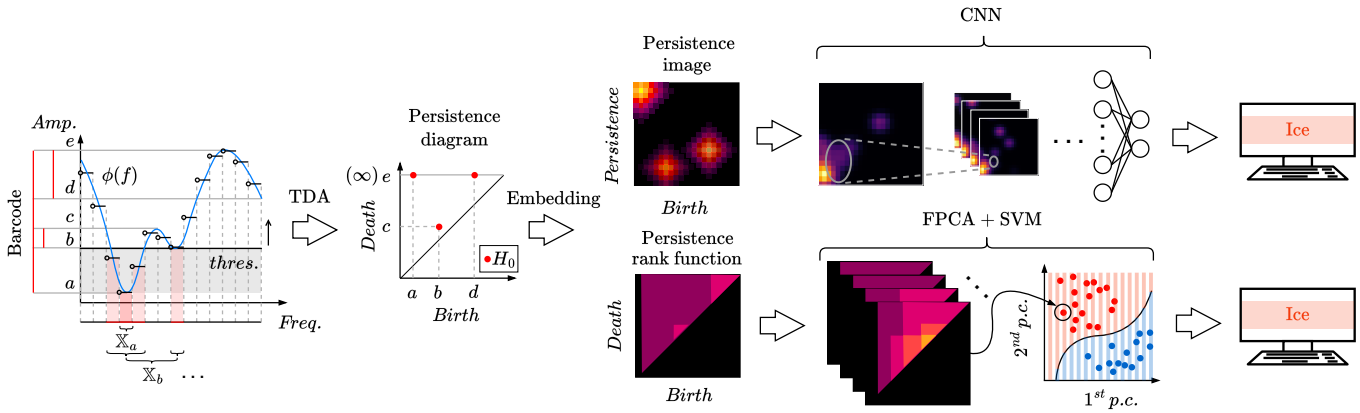


Fig. 4: Proposed workflow.

and reliability. Consequently, a decision was made to develop individual classifiers for each $2\,\mathrm{m/s}$ interval within the wind turbine's operational range considered, spanning from $8$ to $24\,\mathrm{m/s}$. Depending on the real-time mean wind speed, the appropriate classifier among the nine trained ones will be selected and employed for diagnostic purposes.

The $10\,\mathrm{min}$ windows of tower foreaft deflection data used in the training of the classifiers amount to 192 per $2\,\mathrm{m/s}$ wind speed interval, which comprises 4 operation scenarios (normal and ice conditions in 1, 2, and 3 blades), each encompassing 2 turbulence models (NTM and ETM), further divided into 2 wind classes (IB and IC), for which 12 different wind seeds were simulated. For the blade flapwise root bending moments, a total of 288 windows were utilised. This encompasses data from 3 blades for 2 operation scenarios (normal and ice conditions in all 3 blades), which contain the same 2 turbulence models, 2 wind classes and 12 wind seeds.

## IV. Estimation of Tower Foreaft Deflection

TABLE II: Model Constants

| Name | Description |
|---|---|
| $A_r$ | Rotor area |
| $c_t$ | Tower damping |
| $J_g$ | Generator moment of inertia |
| $J_r$ | Rotor moment of inertia |
| $k_t$ | Tower stiffness |
| $m_{t_{eq}}$ | Equivalent tower mass |
| $R$ | Rotor radius |
| $\eta_g$ | Generator efficiency |
| $\mu_d$ | Mechanical losses |
| $\xi_\theta$ | Damping factor of pitch actuator |
| $\rho$ | Air density |
| $\tau_g$ | Generator time constant |
| $\omega_\theta$ | Natural frequency of pitch actuator |

Kalman filtering techniques are widely known for their ability to provide optimal estimates of system states by efficiently combining noisy sensor measurements with dynamic system models. The Unscented Kalman Filter (UKF) is commonly used when dealing with nonlinear system dynamics and non-Gaussian noise distributions, which makes it well-suited to handle nonlinear wind turbine aerodynamics. The procedure is well-known and will not be discussed here; see [16] for an in depth description.

TABLE III: Model Variables

| Name | Description |
|---|---|
| $C_p$ | Power coefficient (look-up table) |
| $C_t$ | Thrust coefficient (look-up table) |
| $F$ | Thrust |
| $n_m$ | Mean wind speed process noise |
| $n_t$ | Wind turbulence process noise |
| $P$ | Power |
| $T_g$ | Generator torque |
| $T_{g_{ref}}$ | Generator torque reference |
| $T_r$ | Aerodynamic torque |
| $v_m$ | Mean wind speed |
| $v_r$ | Wind speed at the rotor |
| $v_t$ | Wind speed turbulent component |
| $x_t$ | Tower foreaft deflection |
| $\theta$ | Collective pitch |
| $\theta_{ref}$ | Collective pitch reference |
| $\lambda$ | Tip-speed ratio |
| $\omega_p$ | Kaimal spectrum peak frequency |
| $\omega_r$ | Rotor speed |

The standard aerodynamic model derived from 1-dimensional momentum theory is used by the UKF instead of complex Blade Element Momentum (BEM) models [17]. While BEM models, used by simulation software like GH Bladed and OpenFAST, accurately model rotor aerodynamics, they prove challenging for control or real-time state estimation due to their complexity and the iterative computations required to determine key variables. The full state-space model as presented in [18] (excluding dynamic inflow corrections) is rewritten hereunder for completeness:

$$\dot{\omega}_r = \frac{(1-\mu_d)\,T_r - T_g}{(J_r + J_g)} \tag{3}$$

$$T_r = \frac{1}{2\,\omega_r}\,\rho\,A_r\,v_r^3\,C_p(\theta,\lambda) \tag{4}$$

$$F = \frac{1}{2}\,\rho\,A_r\,v_r^2\,C_t(\theta,\lambda) \tag{5}$$

$$\lambda = \frac{\omega_r\,R}{v_r} \tag{6}$$

$$v_r = v_m + v_t - \dot{x}_t \tag{7}$$

$$\dot{v}_t = -\omega_p(v_m)\,v_t + \omega_p(v_m)\,n_t \tag{8}$$

$$\dot{v}_m = n_m \tag{9}$$

$$\ddot{x}_t = \frac{1}{m_{t_{eq}}} \left(F - k_t\, x_t - c_t\, \dot{x}_t\right) \tag{10}$$

$$\ddot{\theta} = \omega_\theta^2\, \theta_{ref} - 2\,\omega_\theta\, \xi_\theta\, \dot{\theta} - \omega_\theta^2\, \theta \tag{11}$$

$$\dot{T}_g = \frac{T_{g_{ref}} - T_g}{\tau_g} \tag{12}$$

$$P = \eta_g\, T_g\, \omega_g \tag{13}$$

Check Tables II and III for a description of the constants and variables involved. The power and thrust coefficients ($C_p$ and $C_t$) in Eqs. 4 and 5 are given in the form of a look-up table as a function of the blade pitch angle, $\beta$, and the tip-speed ratio, $\lambda$. The equivalent tower mass in Eq. 10, $m_{t_{eq}}$, is computed assuming that all the mass is located in a point at a height $H$ from the ground as $m_{t_{eq}} = I_t/H^2$, where $I_t$ is the moment of inertia of the tower. The exact value of $I_t$ is unknown, so it is computed by approximating its shape to a hollow cylinder of outer radius $r_t$ and average wall thickness of $0.04\,\mathrm{m}$ as $I_t = m_t/12\left(3\left(r_t^2 + (r_t - 0.04)^2\right) + H^2\right) + m_t(H/2)^2 + m_{nr}H^2$. In the previous expression, $m_t$ refers to the total mass of the tower, $m_t(H/2)^2$ comes from applying the parallel axis theorem to move the axis from the centre of the cylinder to its edge, and $m_{nr}H^2$ accounts for the nacelle and rotor inertia contribution. The tower stiffness, $k_t$, and damping, $c_t$, are found from the expressions below, using the corresponding damping ratio, $\xi_1$, and natural frequency of the $1^{st}$ vibration mode, $\omega_1$, taken from the modal analysis of the $15\,\mathrm{MW}$ wind turbine:

$$\xi_1 = \frac{c_t}{2\, m_{t_{eq}}\, \omega_1}, \qquad \omega_1 = \sqrt{\frac{k_t}{m_{t_{eq}}}} \tag{14}$$

The state, input and measurement vectors are defined as follows:

$$\mathbf{x} = \begin{bmatrix} \omega_r & x_t & \dot{x}_t & \theta & \dot{\theta} & T_g & v_t & v_m \end{bmatrix} \tag{15}$$

$$\mathbf{u} = \begin{bmatrix} \theta_{ref} & T_{g_{ref}} \end{bmatrix} \tag{16}$$

$$\mathbf{y} = \begin{bmatrix} \omega_r & \ddot{x}_t & \theta & P & v_r \end{bmatrix} \tag{17}$$

leading to the following state-space representation with additive noise:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) + \mathbf{n} \tag{18}$$

$$\mathbf{y} = h(\mathbf{x}, \mathbf{u}) + \mathbf{v} \tag{19}$$

In order to define the process noise covariance, $Q$, it is necessary to review the wind model introduced in [18] and [19]. When the wind model is defined in continuous-time (Eqs. 8 and 9), $\mathbf{n}$ can be interpreted as continuous-time white noise corresponding to a Wiener process $\mathbf{w} = \int \mathbf{n}\, dt$, which has a certain incremental covariance $\mathrm{Cov}(\mathbf{w}(t_2) - \mathbf{w}(t_1))$ defined based on a realistic "size" of wind variation. Starting with the mean wind speed in Eq. 9, this stochastic differential equation corresponding to a random walk can be more easily analysed by expressing it in discrete time form:

$$v_m[k] = v_m[k-1] + T_s\, n_m[k-1], \quad n_m \in \mathrm{NID}(0, \sigma_m^2) \tag{20}$$

Here, $n_m$ denotes zero-mean white noise with variance $\sigma_m^2$. Consequently, the variance of $v_m$ increases with each time step, reaching $k\,\sigma_m^2$ after $k$ steps. Therefore, deviations beyond $2\sqrt{k}\,\sigma_m$ are unlikely. Assuming a mean wind speed variation of approximately $2\,\mathrm{m/s}$ over $10\,\mathrm{min}$, $\sigma_m$ can be defined as:

$$\sigma_m = \sqrt{\frac{(2\,\mathrm{m/s})^2}{600\,\mathrm{s}}} \tag{21}$$

The turbulence term of the wind speed has zero mean and variance dependent on the mean wind speed, and is approximated by a first order low pass filter driven by white noise as seen in Eq. 8. The frequency $\omega_p$ is chosen to give the same bandwidth as the peak frequency for the Kaimal spectrum [20]:

$$\omega_p = \frac{v_m\, \pi}{2\, L} \tag{22}$$

where $L = 8.1 \cdot \Lambda_1 = 8.1 \cdot 42 = 340.2$ [9]. Discretizing Eq. 8 using Euler and defining $a = 1 - T_s\omega_p$ yields the standard formula for an infinite-impulse response (IIR) low pass filter:

$$v_t[k] = a\, v_t[k-1] + (1-a)\, n_t[k-1], \quad n_t \in \mathrm{NID}(0, \sigma_t^2) \tag{23}$$

Considering a stationary stochastic process, it is possible to calculate the variance of the previous expression as follows:

$$\mathrm{Var}(v_t) = a^2\, \mathrm{Var}(v_t) + (1-a)^2\, \mathrm{Var}(n_t) \tag{24}$$

Defining $\mathrm{Var}(n_t) = \sigma_t^2$, $\mathrm{Var}(v_t) = \sigma_{v_t}^2$ and rearranging terms yields:

$$\sigma_{v_t}^2 = \frac{(1-a)^2}{1-a^2}\, \sigma_t^2 \tag{25}$$

The turbulence intensity is defined as $t_i = \sigma_{v_t}/v_m$. Isolating $\sigma_{v_t}$ and introducing it in Eq. 25 yields:

$$\sigma_t = t_i\, v_m \sqrt{\frac{1-a^2}{(1-a)^2}} \tag{26}$$

The diagonal terms of the process noise covariance matrix, $Q$, corresponding to the mean and turbulent wind speed components are then the following:

$$\mathrm{Var}\left(\frac{v_m \pi}{2L} n_t\right) = \left(\frac{v_m \pi}{2L}\right)^2 \sigma_t^2 \tag{27}$$

$$\mathrm{Var}\left(n_m\right) = \sigma_m^2 \tag{28}$$

The rest of the diagonal elements have been roughly defined based on the average uncertainty of the model predictions when compared to simulation data. The measurement noise covariance, $R$, has been defined based on the typical errors for the corresponding sensors: $1.7\%$ of the rated rotor speed, $2\%$ of the standard deviation of the tower acceleration, $0.1^\circ$ for the pitch angle, $1\%$ of rated value for the power and $0.5\,\mathrm{m/s}$ for the wind speed at the rotor.

For a reliable implementation, it is necessary to solve the issue of scale in the state variables. The presence of both large and small values (e.g. generator torque compared to pitch angles) produces ill-conditioned covariances which impair filter performance. The Normalised Unscented Kalman Filter (NUKF) introduced in [8] propagates the standard deviation and correlation matrices of the states instead of the covariance, resulting in lower condition numbers when computing the covariance square root and when obtaining the Kalman gain. Consequently, the implementation of the NUKF was deemed

appropriate for this study. Some of the estimated states can be seen in Fig. 5 for a simulation with average wind speed of $14\,\mathrm{m/s}$. The estimation results are relatively noisy due to the large process noise covariance reflecting the inherent uncertainty of the simplified aerodynamic model, in an effort to keep the state estimation realistic.

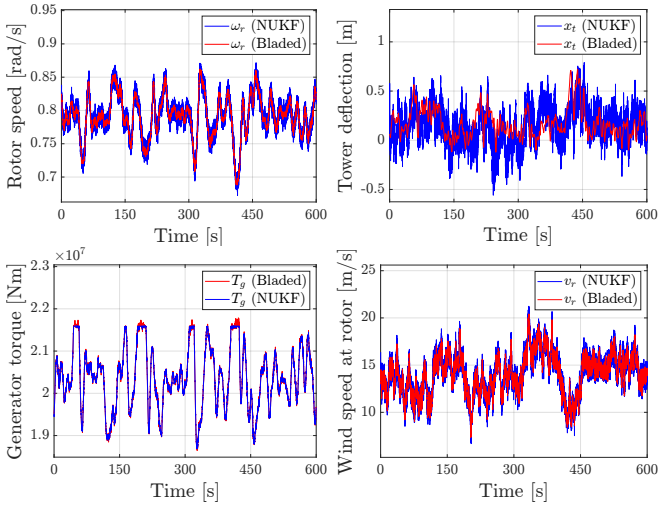


Fig. 5: Some of the NUKF estimated states for a $10\,\mathrm{min}$ simulation with average mean wind speed of $14\,\mathrm{m/s}$, NTM wind conditions and IB wind class. Our variable of interest is the tower foreaft deflection $x_t$ (top, right). Note that noisy Bladed data has been given to the NUKF, but has been plotted without noise for the sake of clarity.

## V. Feature Extraction via Fast Fourier Transform and Sublevel Set Persistent Homology

A well-known method for investigating the dynamic behaviour of the structural components of a wind turbine is through frequency spectral analysis, which allows for the identification and characterisation of the frequency components present in the vibration signals. This analysis aids in identifying the changes of the dominant frequencies due to the presence of ice, even amidst the presence of noise. Computing the discrete Fourier transform of the signals with the FFT algorithm reveals the frequency peaks corresponding to the main modes of vibration of the wind turbine tower and blades. Figure 6 (left) shows a comparison between the 4 operating scenarios considered for a $14\,\mathrm{m/s}$ simulation of tower foreaft deflection with NTM wind conditions and IB wind class. The impact of ice accumulation on one or two blades primarily affects frequencies around 1P ($0.127\,\mathrm{Hz}$ in the simulation shown in Fig. 6). A frequency range spanning from $0.05\,\mathrm{Hz}$ to $0.37\,\mathrm{Hz}$ is selected as main focus to capture the effects of ice, as its impact is minimal for frequencies outside that range. The two asymmetrical ice mass distribution scenarios are easily distinguishable from normal conditions in the tower foreaft deflection spectra, whereas the effect is very subtle when ice is evenly distributed across all three blades. Fortunately, the impact of a symmetrical ice mass distribution is more pronounced in the blade flapwise root bending moments.

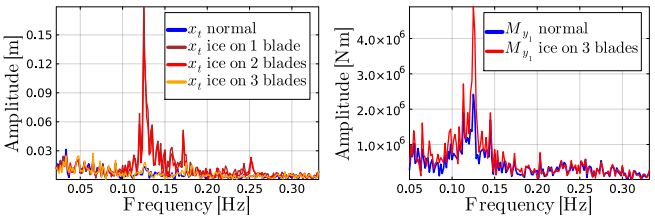


Fig. 6: Tower foreaft deflection (left) and blade flapwise root bending moment spectra (right) for NTM wind conditions and IB wind class showing the four wind turbine operation scenarios considered for a simulation with mean wind speed of $14\,\mathrm{m/s}$.

The spectra presented could be analysed to pinpoint the harmonics experiencing larger alterations caused by the presence of ice, and a threshold could be defined to discern acceptable variations from ice-induced effects. This would be useful in the case of an unbalanced rotor [12], but could not be precise enough to reveal the subtle effect of symmetric ice mass distribution. Alternatively, the spectra could be directly fed to a classifier to solve a supervised learning problem. Instead, it was decided to obtain a more informative object that captures the amplitude of the most prominent harmonics automatically, reducing this way the number of features and making the problem clearer and more separable. To accomplish that, 0-dimensional persistent homology from TDA was used.

Topological data analysis (TDA) is a field of data analysis used to gain insights into complex data sets. It is based onprinciples from algebraic topology, which studies the properties of shapes that are preserved under continuous deformations. TDA aims to understand the shape and structure of data by providing a summary of the connectedness of these data. The focus of this work lies on 0-dimensional topological features, also known as connected components. Connected components provide insights into the excursion sets of single-variable functions, denoted below as $\phi$, which correspond to the spectra under consideration. The sub-level sets will be considered for this work, but a similar analysis could be done with the super-level sets.

Let $\mathbb{X}$ be a topological space in $\mathbb{R}^d$ for some $d \in \mathbb{N}$. For a given level set $r \in \mathbb{R}$, the sub-level set of a tame function $\phi : \mathbb{X} \to \mathbb{R}$ at level $r$ is defined by:

$$\mathbb{X}_r(\phi) := \phi^{-1}\left((-\infty, r]\right) = \{u \in \mathbb{X},\, \phi(u) \leq r\} \qquad (29)$$

and its 0-dimensional homology group is denoted by $H_0(\mathbb{X}_r(\phi))$, which describs the connected components at level $r$, and whose rank $\beta_0$, the $0^{th}$ *Betti number*, is the number of connected components at level $r$. Note that a tame function has excursion sets with finitely many connected components. In this context, $\phi : \mathbb{R} \to \mathbb{R}$; however, the definitions in this section will maintain generality to resemble the notation present in the literature [21].

One important property is that, for all level sets $s, r \in \mathbb{R}$ with $s \leq r$, and for all connected component $C \in H_0(\mathbb{X}_s(\phi))$, there exists a unique connected component $C' \in H_0(\mathbb{X}_r(\phi))$ such that $C \subset C'$. Hence, there is an induced map $\varphi_{s,r} : H_0(\mathbb{X}_s(\phi)) \to H_0(\mathbb{X}_r(\phi))$ which tells us how the connected

components change from level $s$ to $r$. Choosing $n$ different levels $r_1 < r_2 < \cdots < r_n$ we can define a filtration as the nested sequence $H_0(\mathbb{X}_{r_1}(\phi)) \subset H_0(\mathbb{X}_{r_2}(\phi)) \subset \cdots \subset H_0(\mathbb{X}_{r_n}(\phi))$.

Computationally, these filtrations have been defined using cubical complexes, which provide a flexible framework for discretising spaces [22]. A cubical complex is constructed from a finite collection of cubes of varying dimensions, glued together according to certain rules. Consider the topological space $\mathbb{X}$ represented by a cubical complex, where each vertex represents a point in some domain (e.g., a grid in a 2-dimensional plane). To construct a sub-level set filtration using the function $\phi$, which maps the vertices of $\mathbb{X}$ to $\mathbb{R}$, it is necessary to extend it to edges and faces of the complex. For each edge, the function is assigned the value of the higher of its two endpoint vertices. Similarly, for each face, the function is assigned the value of the highest among the values assigned to its defining edges. The filtration is defined by the increasing sequence of values $r_1 < r_2 < \cdots < r_n$, where each $\mathbb{X}_{r_i}$ represents the subcomplex of $\mathbb{X}$ consisting of vertices, edges, and faces for which $\phi$ is less than or equal to $r_i$.

Given $r \in \mathbb{R}$ and $C \in H_0(\mathbb{X}_r(\phi))$, the set of births of connected components is:

$$b(C) = \inf\{s \leq t \,|\, C \in \mathrm{Im}\,(\varphi_{s,t})\} \tag{30}$$

and the set of all deaths is:

$$\begin{aligned} d(C) = \sup\{u \leq r \,|\, \forall C' \in \varphi_{t,u}^{-1}(\varphi_{t,u}(C)), \\ b(C') > b(C)\} \end{aligned} \tag{31}$$

Using the births and deaths of connected components, it is possible to construct a barcode featuring bars represented by intervals $[b(C),\, d(C)]$. Alternatively, by plotting births on the x-axis and deaths on the y-axis, an object called persistence diagram is generated in the form of a scatter plot and denoted as $D(\phi)$ (Fig. 7).

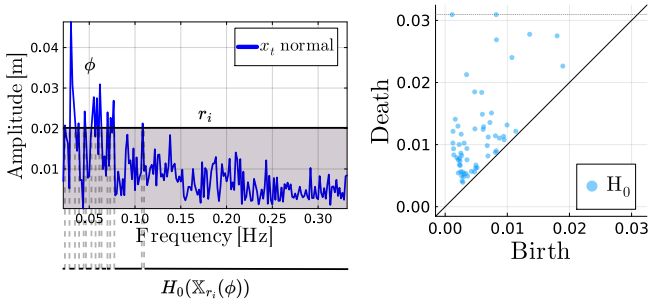


Fig. 7: Filtration snapshot on the normal tower deflection spectra at level $r_i$ (left) and the corresponding persistence diagram (right) for a simulation with mean wind speed of $14\,\mathrm{m/s}$, NTM wind conditions and IB wind class. Note that the connected component with $\infty$ lifetime has not been considered in this example.

There are several metrics that can be defined on the space of persistence diagrams, such as the Bottleneck and Wasserstein distances. One of the most fundamental stability theorems in persistent homology is the Bottleneck stability theorem, which requires the definition of a partial matching between the considered persistence diagrams, denoted as $M : D(f) \leftrightarrow D(g)$. We use the $l_\infty$ metric for the cost associated with a matched pair of points is $(p, q) \in M : ||p-q||_\infty$, while the cost for an unmatched pair is $s \in D(f) \sqcup D(g) : ||s - \bar{s}||_\infty$, where $\bar{s}$ is the projection of $s$ onto the diagonal, typically denoted as $\Delta$. Consequently, the cost of a matching is determined as:

$$\begin{aligned} \mathrm{cost}(M) = \max\{ &\sup_{(p,q)\in M} \left(||p-q||_\infty\right), \\ &\sup_{\substack{s\in D(f)\setminus M \\ s\in D(g)\setminus M \\ \bar{s}\in\Delta}} \left(||s-\bar{s}||_\infty\right)\} \end{aligned} \tag{32}$$

Subsequently, we obtain a distance, also called the *Bottleneck distance* $d_B$, between $D(f)$ and $D(g)$:

$$d_B(D(f), D(g)) = \inf_M \left(\mathrm{cost}(M)\right) \tag{33}$$

The Bottleneck stability theorem is then formulated as:

**Theorem 1.** (Stability) [23] For any tame functions $f, g : \mathbb{X} \to \mathbb{R}$, we have

$$d_B(D(f), D(g)) \leq ||f - g||_\infty.$$

A similar stability theorem can be formulated using the Wasserstein distance, see [24] for the latest results on the matter.

The Bottleneck stability theorem shows that the spectra under consideration is not necessarily more separable in the space of persistence diagrams than in the original space, but automatically extracting this reduced set of 0-dimensional topological features will allow subsequent machine learning techniques to only focus on the changes in the most prominent harmonics, while the others are not as influential, as they may be topological noise.

Handling the space of persistence diagrams poses challenges, particularly in defining the notion of mean (Fréchet mean) due to its computationally intensive nature and the possibility of non-uniqueness [25]. Therefore, it is common practise to embed them into a Hilbert space, where such operations become straightforward. Examples of these embeddings include persistence images or persistence rank functions, as utilised in this paper.

A persistence image [26] takes the persistence diagram in birth/persistence coordinates and treats it as a pixel image, where the value of each pixel depends on the density of topological features in it. Thus, this can be seen as a 2-D vector, which is amenable to standard machine learning algorithms such as CNNs. It is important to remark that direct pixelisation should not be performed for stability purposes of the algorithm. Instead, the image should be smoothed by a Gaussian to ensure sufficient homogeneity.

The persistence rank function [27], represented by Eq. 34, assigns an integer value that corresponds to the number of points within the region $(-\infty, s] \times [r, \infty)$ of the persistence diagram. It is a functional summary which comes equipped with a range of standard statistical techniques such as FPCA.

$$\beta_0(s, r) := \mathrm{rank}(\varphi_{s,r}), \quad \text{for } s < r \tag{34}$$

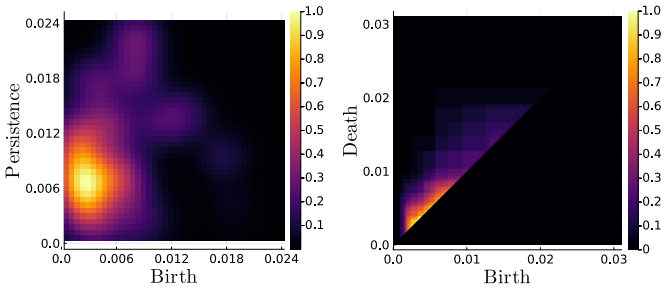


Fig. 8: Persistence image (left) and persistence rank function (right) of the persistence diagram from the scenario presented in Fig. 7.

## VI. Persistence Image Classification with Convolutional Neural Networks

Persistence images of size $40 \times 40$ pixels and standard deviation of the smoothing Gaussian of $\sigma = 0.008$ were generated for each tower deflection persistence diagram out of the 192 available for each wind speed interval of $2\,\mathrm{m/s}$. All persistence images for a certain wind speed interval have the same birth/persistence range, computed based on the connected component with highest persistence in all 192 persistence diagrams. For congruence with the rest of the figures shown in this paper, the results for an average mean wind speed of $14\,\mathrm{m/s}$ are shown in Fig. 9 to exemplify the procedure.

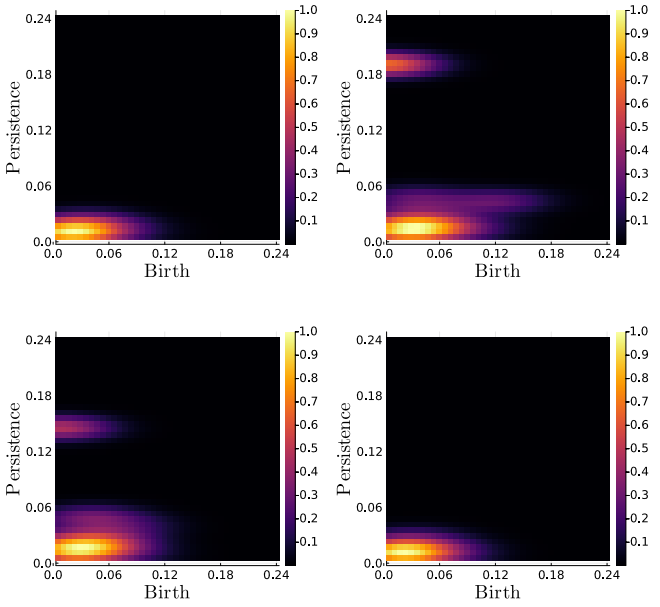


Fig. 9: Persistence images of the tower spectra for the four scenarios considered: normal (top, left), ice in 1 blade (top, right), ice in 2 blades (bottom, left) and ice in 3 blades (bottom, right), for a simulation with mean wind speed of $14\,\mathrm{m/s}$ and NTM, IB wind turbulence.

As expected, the cases with asymmetrical ice mass distribution (top, right and bottom, left) are clearly distinguishable from the normal case (top, left), while the symmetrical ice mass distribution case (bottom, right) is completely indistinguishable from the normal case at this scale. The slight differences between the last two are shown in more detail in Fig. 10, where the original persistence diagrams have been cropped in the range $[0,\, 0.024]$ before embedding them into persistence images, and the standard deviation of the smoothing Gaussian has been reduced to $\sigma = 0.002$.

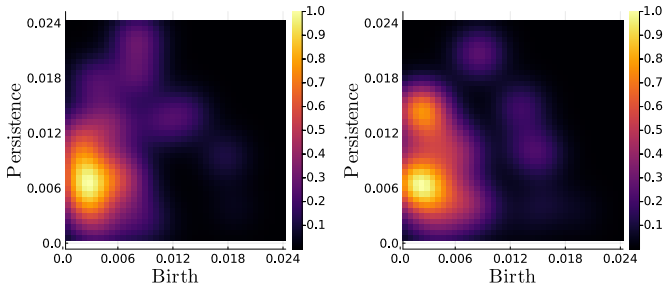


Fig. 10: Zoom in into the persistence images of Fig. 9 for the normal scenario and the case with ice in 3 blades.

Then, Fig. 11 shows an example comparing the normal scenario and the case with symmetrical ice mass distribution across 3 blades, this time using the persistence images derived from flapwise root bending moments, which exhibit a more pronounced dissimilarity compared to those shown in Fig. 9 (top, left and bottom, right) without needing to zoom in as in Fig. 10. As expected, persistence images derived from tower deflection prove more suitable for distinguishing between normal and asymmetric ice mass distributions, whereas those obtained from flapwise root bending moments can be employed to differentiate between normal and symmetric ice mass distributions.

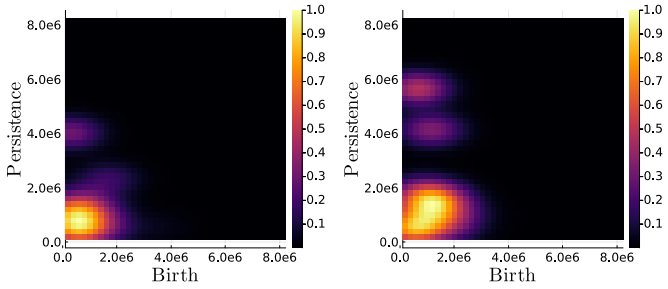


Fig. 11: Persistence images of flapwise root bending moments for the normal scenario and the case with ice across 3 blades for a simulation with mean wind speed of $14\,\mathrm{m/s}$.

CNNs are selected for persistence image classification due to their accuracy and generalisation capabilities when learning hierarchical features directly from raw pixel data, which keeps them as state-of-the-art for image-related tasks.

The CNN used for both tasks has a relatively simple architecture: it begins with a convolutional layer comprising 32

filters, each with a $3 \times 3$ kernel size, using ReLU activation functions. Subsequently, a max-pooling layer with a $2 \times 2$ pool size is applied to reduce the spatial dimensions of the feature maps by selecting the maximum value within each $2 \times 2$ region. This is followed by another convolutional layer with 64 filters and a subsequent max-pooling layer. Then, a flatten layer is introduced to convert the 2D feature maps into a 1D array, preparing them for input into the fully connected layers. Two fully connected layers follow, with 128 and 64 neurons respectively and ReLU activation functions. A dropout regularisation layer with a dropout rate of 0.5 is then added, meaning half of the neurons will be randomly dropped during training, followed by a batch normalisation layer. Finally, an output layer with as many neurons as classes (3 for the first task and 2 for the second task) with softmax activation function outputs probabilities for each class. The model is compiled specifying a categorical cross-entropy loss function and employing the Adam optimiser. The resulting accuracy is $100\%$ for all wind speed intervals and for both tasks. As an example, Fig. 12 shows the learning curves for the second task, classification of normal scenario and the case with symmetrical ice mass distribution in 3 blades, for $14\,\mathrm{m/s}$ average mean wind speed. Refer to Fig. 15 in the Appendix to see the loss curves for the rest of the wind speed intervals within the operating range considered.

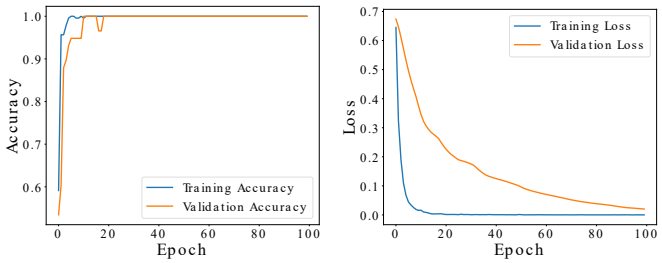


Fig. 12: Model accuracy (left) and loss (right) for the classification problem of persistence images from the normal and ice in 3 blades scenarios, with $14\,\mathrm{m/s}$ average mean wind speed.

## VII. Classification of Persistence Rank Function FPCA Scores Using SVM

Persistence rank functions, as shown in Fig. 13, offer a more precise representation of topological information compared to persistence images, which essentially depict the persistence diagram as seen through a blurry lens. In this analysis, all regions in the persistence rank functions were weighted equally, and a grid size of $153 \times 153$ was defined based on the birth/death range and desired increment in the grid. Analogously to the persistence images, the birth/death range is the same for all persistence rank functions for each wind speed interval, and it is computed based on the connected component with longest lifetime in all 288 persistence diagrams available from root bending moments. Using persistence rank functions enables the application of FPCA to derive scores associated with principal component functions. It is hoped that these scores lie in distinct regions within the principal component space, enabling the use of SVMs for classification. FPCA is well established in the literature, see for example [27] for an introduction of FPCA to persistence rank functions and their use.

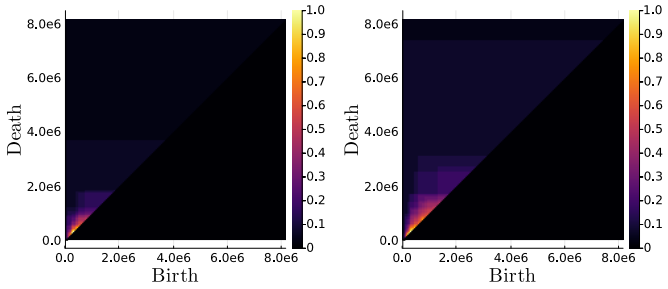


Fig. 13: Persistence rank functions of the blade flapwise root bending moments for the normal scenario and the scenario with ice in 3 blades for the simulation with mean wind speed of $14\,\mathrm{m/s}$ and NTM, IB wind turbulence.

An example of the obtained principal component weight scores for the first two principal components (p.c.) with $14\,\mathrm{m/s}$ average mean wind speed are plotted in the principal component space spanned by the first two p.c. in Fig. 14, and an SVM with radial basis function kernel and default hyperparameters, $C = 1$ and $\gamma = 1/(\text{no. of p.c.} \times \text{p.c. variance})$, is trained with these scores to define the decision boundary seen in the figure. Refer to Fig. 16 in the Appendix to see the decision boundaries for the rest of the wind speed intervals within the operating range considered. The FPCA scores were calculated for both NTM and ETM turbulence models, as well as for IB and IC wind classes. Examining the second column of Table IV reveals that relying solely on the first two principal components fails to mitigate false positives, which are unacceptable in an industrial setting. Consequently, increasing the number of principal components used to characterise the persistence rank functions becomes necessary. The resulting classification accuracies using 6 and 12 principal components are shown in the third and fourth columns of Table IV. Achieving $100\%$ classification accuracy requires a minimum of 12 principal components. As computation time is not a concern, this dimensionality is considered optimal.

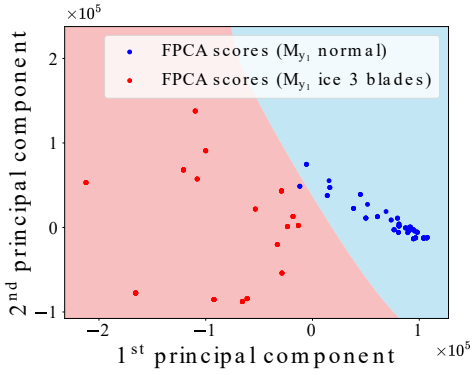


Fig. 14: FPCA scores for the simulation with mean wind speed of $14\,\mathrm{m/s}$.

TABLE IV: Accuracy FPCA+SVM vs. CNN

| Wind speed | 2D FPCA + SVM | 6D FPCA + SVM | 12D FPCA + SVM | CNN |
|---|---|---|---|---|
| 8 m/s | 96.6% | 100% | 100% | 100% |
| 10 m/s | 96.6% | 100% | | |
| 12 m/s | 96.6% | 100% | | |
| 14 m/s | 97.4% | 100% | | |
| 16 m/s | 98.3% | 100% | | |
| 18 m/s | 91.7% | 99.5% | | |
| 20 m/s | 89.1% | 90.9% | | |
| 22 m/s | 81.3% | 90.9% | | |
| 24 m/s | 82.6% | 94.3% | | |

## VIII. Conclusion

In this work, signals from tower foreaft deflection and blade flapwise root bending moments are selected as the main variables affected by the accumulation of ice in the blades, and are used for its diagnosis. Data is simulated using a simulation model provided by Emerson in the software GH Bladed, and different wind conditions were used to ensure realistic results: normal turbulence model (NTM), extreme turbulence model (ETM), and wind classes IB and IC. Tower deflection is used to distinguish asymmetrical ice mass distribution in one or two blades form normal operating conditions, as it provokes noticeable oscillations in this variable due to the mass imbalance, while the presence of symmetrical ice mass distribution in the 3 blades can only be perceived in the flapwise root bending moments. A normalised unscented Kalman filter (NUKF) is used together with a standard aerodynamic and mechanical model to estimate tower deflection, while measurements from root bending moments are assumed to be directly obtained from blade root sensors. The signals are subsequently analysed in the frequency domain by applying the fast Fourier transform (FFT), and 0-dimensional persistent homology from topological data analysis (TDA) is used to analyse the lifetime of the sub-level sets of the spectra from the mentioned four different classes. The objective is to use the 0-dimensional topological features to obtain a clearer supervised learning classification problem than when dealing with the spectra directly. This method encodes the sub-level set information in persistence diagrams, which are embedded in two different persistence summaries: persistence images and persistence rank functions. Each embedding allows for the application of a specific machine learning method, whether it involves direct classification of persistence images using convolutional neural networks (CNNs), or further feature extraction of persistence rank functions through functional principal component analysis (FPCA) to generate a set of scores, that can then be used to define a decision boundary using support vector machines (SVMs). The comparison of the two methods show that both can be tuned to result in a precise and reliable classification problem. Future work will introduce specific weighting functions to enhance the informativeness of these topological summaries, as well as compare them to other alternatives such as persistence landscapes or accumulated persistence functions.

## Acknowledgments

We thank Emerson for sharing data from a 15 MW wind turbine GH Bladed model so that this work could be accomplished.

## References

[1] R. Cattin, "Icing of wind turbines: Vindforsk projects, a survey of the development and research needs, elforsk report 12: 13," 2013.
[2] M. Kreutz, A. A. Alla, A. Eisenstadt, M. Freitag, and K.-D. Thoben, "Ice detection on rotor blades of wind turbines using rgb images and convolutional neural networks," *Procedia CIRP*, vol. 93, pp. 1292–1297, 2020.
[3] L. Zhang, K. Liu, Y. Wang, and Z. B. Omariba, "Ice detection model of wind turbine blades based on random forest classifier," *Energies*, vol. 11, no. 10, p. 2548, 2018.
[4] G. A. Skrimpas, K. Kleani, N. Mijatovic, C. W. Sweeney, B. B. Jensen, and J. Holboell, "Detection of icing on wind turbine blades by means of vibration and power curve analysis," *Wind Energy*, vol. 19, no. 10, pp. 1819–1832, 2016.
[5] X. Wang, Z. Zheng, G. Jiang, Q. He, and P. Xie, "Detecting wind turbine blade icing with a multiscale long short-term memory network," *Energies*, vol. 15, no. 8, p. 2864, 2022.
[6] X. Cheng, F. Shi, Y. Liu, X. Liu, and L. Huang, "Wind turbine blade icing detection: A federated learning approach," *Energy*, vol. 254, p. 124441, 2022.
[7] P. Egedal, "Leveraging insight from operational data to optimize performance in cold climates," in *Winterwind 2018, Åre, Sweden, 5-7 Febuary 2018*, 2018.
[8] H. A. Krog and J. Jäschke, "The simple solution for nonlinear state estimation of ill-conditioned systems: The normalized unscented kalman filter," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 5951–5956, 2023.
[9] I. IEC, "61400-1: Wind turbines part 1: Design requirements," *International Electrotechnical Commission*, vol. 177, 2005.
[10] G. Lloyd and G. Hamburg, "Guideline for the certification of wind turbines," *July 1st*, 2010.
[11] S. Gantasala, "Detection of blade icing and its influence on wind turbine vibrations," Ph.D. dissertation, Luleå University of Technology, 2019.
[12] T. Knudsen, T. Bak, and S. Tabatabaeipour, "Detection of excessive wind turbine tower oscillations fore-aft and sideways," in *2012 American Control Conference (ACC)*. IEEE, 2012, pp. 5795–5800.
[13] M. Čufar, "Ripserer.jl: flexible and efficient persistent homology computation in julia," *Journal of Open Source Software*, vol. 5, no. 54, p. 2614, 2020. [Online]. Available: https://doi.org/10.21105/joss.02614
[14] M. Čufar, "Persistencediagrams.jl," https://github.com/mtsch/PersistenceDiagrams.jl, 2020.
[15] Y. B. Bleile, "Discretepersistenthomologytransform.jl," https://github.com/yossibokorbleile/DiscretePersistentHomologyTransform.jl, 2020.
[16] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.
[17] M. Hansen, *Aerodynamics of wind turbines*. Routledge, 2015.
[18] T. Knudsen and T. Bak, "Simple model for describing and estimating wind turbine dynamic inflow," in *2013 American Control Conference*. IEEE, 2013, pp. 640–646.
[19] T. Knudsen, T. Bak, and M. Soltani, "Prediction models for wind speed at turbine locations in a wind farm," *Wind Energy*, vol. 14, no. 7, pp. 877–894, 2011.
[20] T. Burton, N. Jenkins, D. Sharpe, and E. Bossanyi, *Wind energy handbook*. John Wiley & Sons, 2011.
[21] P. Bendich, H. Edelsbrunner, D. Morozov, and A. Patel, "Homology and robustness of level and interlevel sets," *Homology, Homotopy and Applications*, vol. 15, no. 1, pp. 51–72, 2013.
[22] T. Kaczynski, K. M. Mischaikow, and M. Mrozek, *Computational homology*. Springer, 2004, vol. 157, no. 2.
[23] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer, "Stability of persistence diagrams," in *Proceedings of the twenty-first annual symposium on Computational geometry*, 2005, pp. 263–271.
[24] P. Skraba and K. Turner, "Wasserstein stability for persistence diagrams," *arXiv preprint arXiv:2006.16824*, 2020.
[25] K. Turner, Y. Mileyko, S. Mukherjee, and J. Harer, "Fréchet means for distributions of persistence diagrams," *Discrete & Computational Geometry*, vol. 52, pp. 44–70, 2014.
[26] H. Adams, T. Emerson, M. Kirby, R. Neville, C. Peterson, P. Shipman, S. Chepushtanova, E. Hanson, F. Motta, and L. Ziegelmeier, "Persistence images: A stable vector representation of persistent homology," *Journal of Machine Learning Research*, vol. 18, no. 8, pp. 1–35, 2017.

[27] V. Robins and K. Turner, "Principal component analysis of persistent homology rank functions with case studies of spatial point patterns, sphere packing and colloids," *Physica D: Nonlinear Phenomena*, vol. 334, pp. 99–117, 2016.

## APPENDIX
## RESULTS FOR ALL WIND SPEED INTERVALS

The results obtained for all wind speed intervals within the operating range considered are shown hereunder. Figure 15 shows the CNN loss curves, and Fig. 16 depicts the decision regions defined by SVMs for the persistence rank function FPCA scores of the first two principal components.

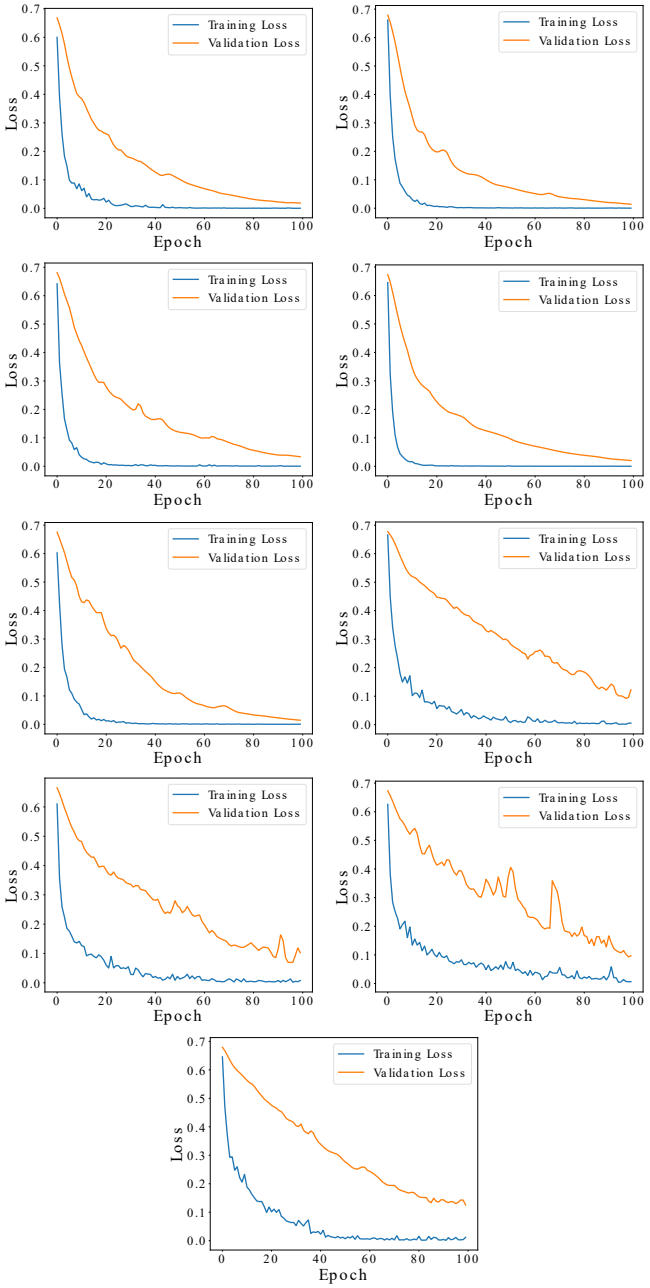


Fig. 15: CNN loss function for the classification of persistence images from blade flapwise root bending moments into normal scenario and the scenario with ice in 3 blades, for wind speeds ranging from $8\,\mathrm{m/s}$ to $24\,\mathrm{m/s}$ in intervals of $2\,\mathrm{m/s}$.

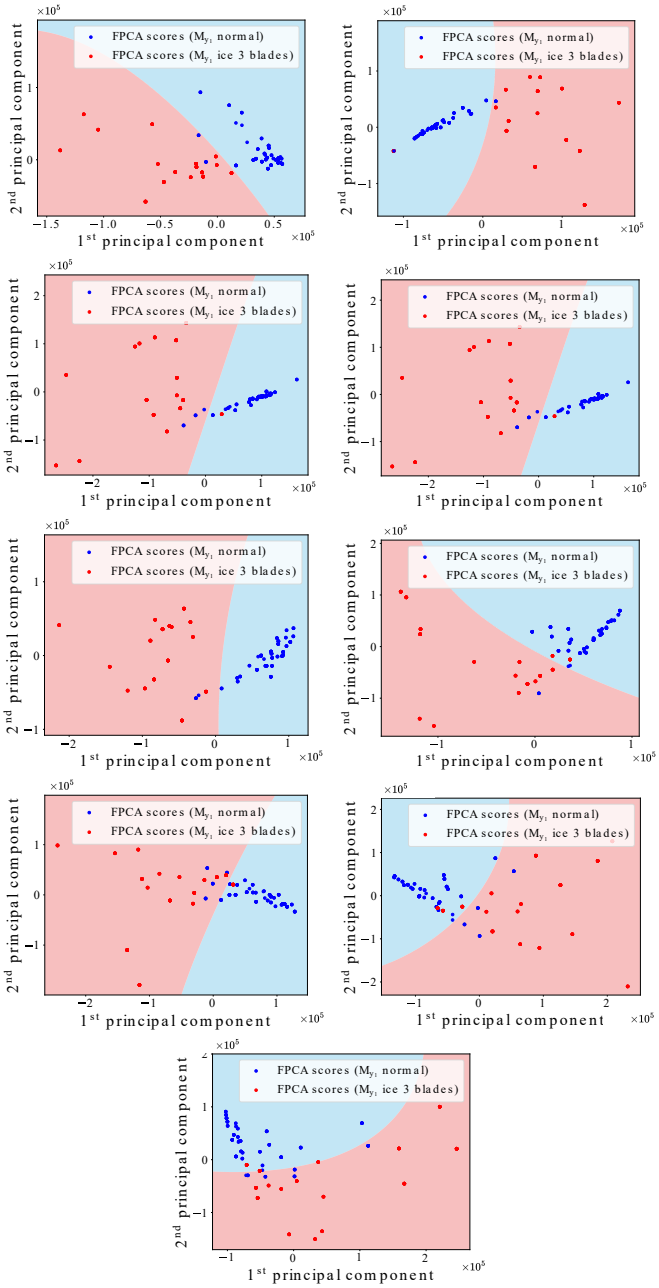


Fig. 16: SVM decision boundaries of persistent rank function FPCA scores for the first two principal components with wind speeds ranging from $8\,\mathrm{m/s}$ to $24\,\mathrm{m/s}$ in intervals of $2\,\mathrm{m/s}$.

**Álvaro M. Gómez** received the M.Sc. in Control and Automation from Aalborg University, Aalborg, Denmark in 2022. He is currently pursuing a Ph.D. within the Department of Electronic Systems at the same university, and he is part of the Learning and Decisions Lab research group. His research focuses on the integration of physical knowledge of dynamical systems with data-driven techniques to overcome the limitations inherent in each approach individually. He seeks to develop new techniques for the accurate and efficient detection of anomalies and cyber-attacks in wind farms.

**Thomas Haugaard** received his M.Sc. in Control and Automation from Aalborg University, Aalborg, Denmark in 2022. He is currently working for Emerson Power and Water Solutions as a wind turbine control engineer. His work is focused on developing control solutions by optimising algorithms and expanding functionality, including developing fault detection techniques. He is committed to pushing the boundaries of wind turbine technology.

**Oier Ajenjo de Torres** graduated with a M.Sc. in Control and Automation from Aalborg University, Denmark, in 2022. At Nordea, he is currently employed as a Technology Graduate, concentrating on advancing artificial intelligence and cloud systems to aid in adopting new technologies in the finance industry. With extensive experience in computer engineering, control and automation, and a passion for state-of-the-art technologies, he is committed to pushing forward the fusion of finance and technology, guaranteeing creative answers for the changing demands of the sector.

**Torben Knudsen** received his Ph.D. degree in 1990 at the Institute for Mathematical Modelling at Aalborg University, Aalborg, Denmark. Since 2006 he has been an associate professor in the department of Electronic Systems at the same university, in the section of Automation and Control. In 1986 he was a research assistant at Wind Energy and Atmospheric Physics Department at Risø National Laboratory. From 1987 to 2002 he was at the Department of Electronic Systems both as assistant and associate professor. From 2002 to 2006 he was working as a control development engineer with Vestas Wind Systems A/S R&D. His main research areas are stochastic systems, state estimation, system identification and control. Concerning applications, he has been working mostly with wind energy in various project on single turbine control, floating turbines and wind farm operation.

**Yossi Bokor Bleile** completed his Joint PhD between the Mathematical Sciences Institute, Australian National University and the School of Mathematics and Statistics, University of Sydney, in Geometric and Topological Data Analysis, in 2022. He is currently a Post Doc at the Department of Mathematical Sciences, Aalborg University, Aalborg, Denmark, and a member of the Danish Data Science Academy Young Academy Panel (YAP). He is interested in object oriented data analysis, in particular developing geometric and topological tools to address object classification, object reconstruction, and understanding the relationship between morphological features and object properties.

**Rafał Wiśniewski** obtained his first Ph.D. degree in 1997 in the Department of Control Engineering, and his second Ph.D. degree in 2005 in the Department of Mathematical Sciences, both at Aalborg University, Aalborg, Denmark. He is a professor in the Department of Electronic Systems at the same university, in the section of Automation and Control, and the Head of Doctoral Program in Electrical and Electronic Engineering. His research objectives are to establish mathematical formalism for analysis of cyber-physical systems, and to develop numerical methods for solving fundamental control problems for non-linear systems: stability, robustness, and safety. He has over 50 peer reviewed international journal papers, and over 100 peer reviewed conference papers. He received the Aalborg City Springtime Award in 1999 for his contributions of the Satellite Team at Aalborg University to advance the status of society and the recognition of the city. He secured major grants with him as a principal investigator, such as CodeMe, EDGE, Smart & Cool, Wind Turbine Control Program and Hybrid Control.