Neural Computing and Applications
https://doi.org/10.1007/s00521-021-06288-w

ORIGINAL ARTICLE

# The stochastic aeroelastic response analysis of helicopter rotors using deep and shallow machine learning

Tanmoy Chatterjee[1] · Aniekan Essien[2] · Ranjan Ganguli[3] · Michael I. Friswell[1]

Received: 5 December 2020 / Accepted: 26 June 2021
© The Author(s) 2021

## Abstract

This paper addresses the influence of manufacturing variability of a helicopter rotor blade on its aeroelastic responses. An aeroelastic analysis using finite elements in spatial and temporal domains is used to compute the helicopter rotor frequencies, vibratory hub loads, power required and stability in forward flight. The novelty of the work lies in the application of advanced data-driven machine learning (ML) techniques, such as convolution neural networks (CNN), multi-layer perceptron (MLP), random forests, support vector machines and adaptive Gaussian process (GP) for capturing the nonlinear responses of these complex spatio-temporal models to develop an efficient physics-informed ML framework for stochastic rotor analysis. Thus, the work is of practical significance as (i) it accounts for manufacturing uncertainties, (ii) accurately quantifies their effects on nonlinear response of rotor blade and (iii) makes the computationally expensive simulations viable by the use of ML. A rigorous performance assessment of the aforementioned approaches is presented by demonstrating validation on the training dataset and prediction on the test dataset. The contribution of the study lies in the following findings: (i) The uncertainty in composite material and geometric properties can lead to significant variations in the rotor aeroelastic responses and thereby highlighting that the consideration of manufacturing variability in analyzing helicopter rotors is crucial for assessing their behaviour in real-life scenarios. (ii) Precisely, the substantial effect of uncertainty has been observed on the six vibratory hub loads and the damping with the highest impact on the yawing hub moment. Therefore, sufficient factor of safety should be considered in the design to alleviate the effects of perturbation in the simulation results. (iii) Although advanced ML techniques are harder to train, the optimal model configuration is capable of approximating the nonlinear response trends accurately. GP and CNN followed by MLP achieved satisfactory performance. Excellent accuracy achieved by the above ML techniques demonstrates their potential for application in the optimization of rotors under uncertainty.

**Keywords** Helicopter rotor · Aeroelastic · Stochastic · Machine learning

## 1 Introduction

Helicopters experience high level of vibrations compared to other flight vehicles due to a significantly higher degree of aeroelastic interaction and rapidly rotating flexible blades [21]. The vibratory loads in helicopters typically emanate from the main rotor and can result in fatigue damage of important structural components, cause human discomfort and reduce the efficacy of weapon systems. Therefore, considerable research has been directed towards accurate modelling of helicopter rotor blades [35, 48]. Rotorcraft analysis is typically conducted using comprehensive codes [22]. These codes are needed to provide

✉ Tanmoy Chatterjee
tanmoy.chatterjee@swansea.ac.uk

[1] College of Engineering, Swansea University, Bay Campus, SA1 8EN Swansea, United Kingdom

[2] Information Systems (Management), University of Sussex Business School, Sussex House, Falmer Brighton BN1 9RH, United Kingdom

[3] Department of Aerospace Engineering, Indian Institute of Science, Bangalore 560012, India

Published online: 17 July 2021

aeromechanics predictions for helicopter properties such as performance, vibration and aeroelastic stability.

Composite materials have been a natural choice for helicopter rotor blades owing to their superior strength, high stiffness-to-weight ratios and other properties which can be tailored based on requirements. In this context, the underlying assumption of multiple existing studies is that the aeroelastic response of composite helicopter rotor blade corresponding to deterministic physical (input) parameters, replicates the actual behaviour. This assumption often proves to be invalid especially for practical industrial applications where the presence of uncertainty is inevitable. Therefore, besides enhancing the fidelity of the deterministic model of composite rotor blades, the quantification of the response variation because of manufacturing anomalies is equally important, if not more so, for a more realistic description of the system behaviour. Moreover, the effect of manufacturing variability on the aeroelastic response may be aggravated due to the inherent nonlinearities and structural and aerodynamic interactions.

Consequently, active research has been carried out over the years to quantify the influence of uncertainties on the dynamic response of aerospace structures. The first comprehensive review of uncertainty quantification (UQ) in aeroelasticity can be found in [43]. For a more recent survey on the same topic, [5, 12] is recommended. However, the literature seems to be quite scarce when it comes to specific works related to the intersection of aeroelasticity, UQ, composite modelling and helicopter rotor blades. The honorable mentions of existing works falling under the above multi-disciplinary intersection are as follows:

For the first time, the influence of composite material uncertainty on the aeroelastic properties of a helicopter rotor was investigated in [39]. In particular, they studied the effect of aleatoric uncertainties on the aeroelastic response of the helicopter rotor and vibratory hub loads. Manufacturing constraints were introduced within the multidisciplinary rotor blade optimization framework in [32]. In doing so, durability and fatigue analysis were performed coupled with a probabilistic robust design methodology to reduce the effects of material, geometry (particularly, shape) and loading uncertainties on the rotor blade structural performance. The influence of spatially varying material properties were studied on the aeroelastic code predictions (e.g., rotating natural frequencies, vibratory hub loads, etc.) of composite helicopter rotors in [40]. A stochastic spectral method was employed combining Karhunen–Loéve expansion and high dimensional model representation to reduce the computational cost. Epistemic uncertainty modelling was performed in [6], illustrating high sensitivity of vibratory loads on the optimal design of inflow models. The authors went on to demonstrate the high sensitivity, such that the optimal configuration of one inflow model performed worse than the baseline design when evaluated with a different inflow model. The influence of material and manufacturing uncertainties of a composite UH-60A helicopter rotor blade model were propagated to the beam properties, the rotating natural frequencies, the aeroelastic response, and vibratory loads in hover and forward flight [44]. A micro-mechanical stochastic approach was undertaken by varying the fiber orientations of the box-spar of high-fidelity rotor-blade models. An experimental technique was recently devised for flutter speed UQ as a stochastic structural modification problem considering manufacturing tolerances, damage and degradation [2].

The common aspect and key take-away from findings of the above articles is that perturbations in the material and geometric parameters of a composite helicopter rotor can lead to significant fluctuations in the aeroelastic dynamic response, thereby accentuating the requirement for stochastic analysis. In this context, UQ has retained its popularity since past few decades. Despite its usefulness, it is computationally expensive to implement in large-scale systems [53]. Therefore, cost-effective non-intrusive UQ tools can be useful for analyzing such computationally demanding systems as they entail detailed numerical models and sophisticated deterministic solvers in which one cannot readily modify the existing framework to set up the necessary propagation tools. This is altogether more relevant for rotor analysis as the governing equations of rotorcraft aeroelasticity are typically nonlinear and any alteration of the rotorcraft analysis software needs domain specialists.

Monte Carlo Simulation (MCS) is the most widely used and simplest approach for stochastic response analysis [41]. However, MCS requires large number of simulations and thus, proves to be inefficient for large-scale detailed models. Substantial research has been carried out to improve the computational framework of MCS. In contrast to MCS and its variants, which are essentially sampling based approaches, non-sampling-based techniques, such as surrogate models are computationally viable alternative to the former [31]. These models map the input-output relationship and approximate the functional space with the help of small number of actual physics-based high-fidelity simulations. This reduces the computational effort significantly. Some recent research has used surrogate modeling for aerospace analysis and design. Batrakov et al. [3] performed optimization for the rear fuselage of a helicopter using genetic algorithms and Kriging surrogate models. Lu et al. [34] used a Kriging surrogate model of the objective function along with genetic algorithm to reduce the adverse effects of aerodynamic interactions on UH-60 type fuselage of a helicopter. Kontogiannis et al. [27] used Kriging and co-Kriging based multifidelity surrogates for

aerodynamic optimization. Extensive review of surrogate models can be found in [15, 19].

Although machine learning (for example, radial basis function neural network [29], recurrent neural network and multi-layer perceptron [30]) has been employed in solving inverse problems (for example, structural health monitoring, damage detection and model updating) for rotor blade applications previously, however, we observed that the literature is scarce when it comes to the application of ML for forward stochastic aeroelastic response analysis of rotors. In this context, it is worth mentioning a recent work which has employed deep learning to emulate and extrapolate from the limited experimental responses of rotorcraft available as raw sensor (accelerometer) data and create a 'virtual sensor' for better understanding of their vibration behaviour [36]. A data-driven framework was proposed to develop safety-based diagnostics for rotorcrafts and to define the process of selecting a single, airworthy ML-based diagnostic classifier that replaces a suite of fielded condition indicators (CI) [54]. A high-performance parallel computing framework for deep neural network (DNN) hyperparameter search using evolutionary optimization was proposed for nonlinear high-dimensional multivariate regression problems for condition monitoring of rotorcrafts [17]. The developed DNN models were capable of mapping existing CI to helicopter oil cooler vibration spectra and thereby infer the quality of the internal bearing faults [18]. The above works are a part of improving the Health and Usage Monitoring Systems (HUMS) in rotorcrafts via ML, initiated by the US Army Aviation Engineering Directorate (AED) [55]. HUMS evaluate CI to quantify the rotorcraft health state from operational flight data collected from on-board sensors. It is to be noted that the above works are data-driven ML-based rotorcraft operational analyses and did not account for (i) physics-based modelling or (ii) any form of uncertainties. Drawing motivation from the above works, this work builds upon generating ML models on limited and expensive synthetic rotor response datasets resulting from high-fidelity physics-based models for accurate and efficient UQ.

Specifically, the following points have motivated our research: (i) generation of response data by solving detailed physics-based models is computational expensive and (ii) input-response relationship is strongly nonlinear. While (i) can be addressed by conventional surrogate models, we investigate various specialized ML techniques and utilize their multi-layered architecture in this work for ensuring satisfactory approximation accuracy (point (ii)). To be precise, our work attempts to improve upon the accuracy aspect by metamodeling of the stochastic rotor responses.

To the best of the authors' knowledge, this is the first application of advanced machine learning-driven stochastic aeroelastic analysis of helicopter rotor blades. The rest of the paper is organized in the following sequence. The aeroelastic analysis is discussed briefly in Sect. 2. The stochastic response by ML is illustrated in Sect. 3. In Sect. 4, the numerical study is undertaken and the results are interpreted. Finally, the work is summarized in Sect. 5.

## 2 Aeroelastic analysis

For realistic prediction of helicopter vibratory hub loads, an aeroelastic analysis is required. A comprehensive aeroelastic analyses software has been created to address this issue [21]. This aeroelastic analysis is briefly elucidated below. The equations in this section are adopted from [21] and are provided here for completeness.

### 2.1 Governing equations of motion

The helicopter is modeled as a nonlinear model of multiple elastic rotor blades coupled to a six-degree-of-freedom rigid fuselage. Each blade displays flap (out-of-plane) bending ($w$), lag (in-plane) bending ($v$), elastic twist (torsion) ($\phi$) and axial displacement ($u$) as shown in Fig. 1. The equations of motion are derived using the generalized Hamilton's principle developed for non-conservative systems:

$$\delta\Pi = \int_{\psi_1}^{\psi_2} (\delta U - \delta T - \delta W) d\psi = 0 \tag{1}$$

where $\delta U$, $\delta T$ and $\delta W$ represent the virtual variations of strain energy, kinetic energy and virtual work performed by an external force, respectively, and $\delta\Pi$ indicates the total

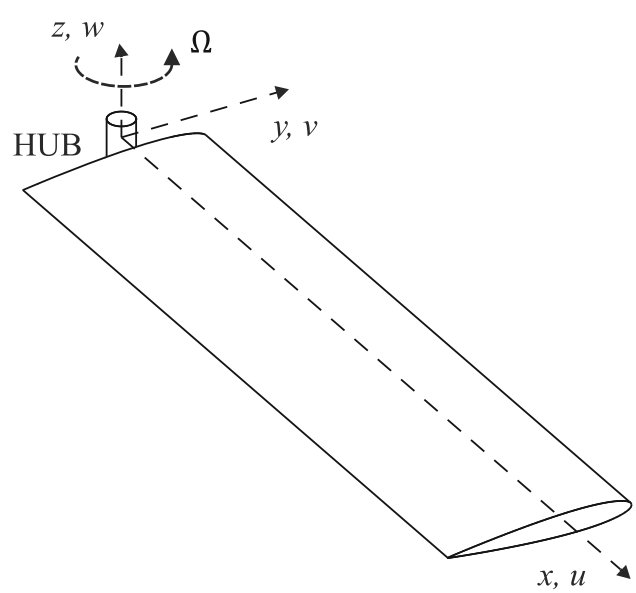


**Fig. 1** Elastic rotor blade

potential of the system. The $\delta U$ and $\delta T$ are derived using the Hodges and Dowell approach and incorporate a moderate deflection theory [21]. The external aerodynamic forces acting on the rotor blade add to the virtual work variational $\delta W$. The aeroelastic analysis applied in this paper considers aerodynamic forces and moments which are calculated using free wake analysis, incorporate a reverse flow model and address time domain unsteady aerodynamics.

## 2.2 Finite element-spatial discretization

The governing equations of motion are converted to discrete form using finite element (FE) analysis. This analysis is valid for nonuniform blade properties. Once discretized, equation (1) is expressed as:

$$\delta\Pi = \int_0^{2\pi} \sum_{i=1}^{N} (\delta U_i - \delta T_i - \delta W_i) d\psi = 0 \tag{2}$$

The beam is divided into $N$ spatial finite elements. Each of the $N$ finite elements incorporates fifteen degrees of freedom. These fifteen degrees of freedom incorporate six degrees of freedom at each boundary node (axial and torsion displacement, flap and lag bending displacement, flap and lag bending slope) and two internal nodes for axial displacement and one internal node for torsion displacement. These degrees of freedom correspond to cubic variations in axial elastic and (flap and lag) bending deflections, and quadratic variation in elastic torsion. Between the elements, there is continuity of slope and displacement for flap and lag bending deflections and continuity of displacements for elastic twist and axial deflections. This FE guarantees physically consistent linear variations of bending moments and torsion moments and quadratic variations of axial force inside the elements. The shape functions used here are Hermite polynomials for lag and flap bending and Lagrange polynomials for axial and torsion deflection and are given in [7]. In this paper, cantilever boundary conditions are considered and the rows and columns corresponding to the root node in the global mass, stiffness, damping matrices and the force vector are discarded. For the numerical results, a non-uniform mesh with thirteen elements is used.

Substituting $\mathbf{u} = \mathbf{H}\mathbf{q}$ ( $\mathbf{H}$ is the shape function matrix) into the expression for Hamilton's principle yields:

$$\int_0^{2\pi} \delta\mathbf{q}^{\mathrm{T}} \left( \mathbf{M}\ddot{\mathbf{q}}(\psi) + \mathbf{C}\dot{\mathbf{q}}(\psi) + \mathbf{K}\mathbf{q}(\psi) - \mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}, \psi) \right) \mathrm{d}\psi = 0 \tag{3}$$

The space functionality is thus eliminated by applying FE discretization and the governing partial differential equations are reduced to ordinary differential equations.

## 2.3 Normal mode transformation

Each rotor blade is modeled using FE equations. These equations are transformed into normal mode space to facilitate the computationally efficient solution of blade response. The displacements are enunciated with respect to normal modes as $\mathbf{q} = \mathbf{\Phi}\mathbf{p}$. Substituting $\mathbf{q} = \mathbf{\Phi}\mathbf{p}$ in Eq. (3) yields the equations in normal mode coordinates:

$$\int_0^{2\pi} \delta\mathbf{p}^{\mathrm{T}} \left( \bar{\mathbf{M}}\ddot{\mathbf{p}}(\psi) + \bar{\mathbf{C}}\dot{\mathbf{p}}(\psi) + \bar{\mathbf{K}}\mathbf{p}(\psi) - \bar{\mathbf{F}}(\mathbf{p}, \dot{\mathbf{p}}, \psi) \right) \mathrm{d}\psi = 0 \tag{4}$$

where the mass, stiffness and damping matrices and the force vector in the normal mode space are expressed as $\bar{\mathbf{M}} = \Phi^T\mathbf{M}\Phi$, $\bar{\mathbf{C}} = \Phi^T\mathbf{C}\Phi$, $\bar{\mathbf{K}} = \Phi^T\mathbf{K}\Phi$ and $\bar{\mathbf{F}} = \Phi^T\mathbf{F}$ , respectively. Integration of Eq. (4) by parts yields:

$$\int_0^{2\pi} \begin{Bmatrix} \delta\mathbf{p} \\ \delta\dot{\mathbf{p}} \end{Bmatrix} \begin{Bmatrix} \bar{\mathbf{F}} - \bar{\mathbf{C}}\dot{\mathbf{p}} - \bar{\mathbf{K}}\mathbf{p} \\ \bar{\mathbf{M}}\dot{\mathbf{p}} \end{Bmatrix} d\psi = \begin{Bmatrix} \delta\mathbf{p} \\ \delta\dot{\mathbf{p}} \end{Bmatrix} \begin{Bmatrix} \bar{\mathbf{M}}\dot{\mathbf{p}} \\ 0 \end{Bmatrix} \Bigg|_0^{2\pi} \tag{5}$$

The right hand side of the aforementioned equation vanishes due to the periodic nature of the rotor steady state response. Consequently, Eq. (5) generated the system of first order differential equations:

$$\int_0^{2\pi} \delta\mathbf{y}^{\mathrm{T}}\mathbf{Q}\, \mathrm{d}\psi = 0$$

$$\text{where} \quad \mathbf{y} = \begin{Bmatrix} \mathbf{p} \\ \underline{\mathbf{p}} \end{Bmatrix} \quad \text{and} \quad \mathbf{Q} = \begin{Bmatrix} \overline{\mathbf{F}} - \overline{\mathbf{C}}\underline{\mathbf{p}} - \overline{\mathbf{K}}\mathbf{p} \\ \overline{\mathbf{M}}\underline{\mathbf{p}} \end{Bmatrix} \tag{6}$$

For the numerical results, four flap, three lag, two torsion and one axial mode are used.

## 2.4 Finite element-temporal discretization

The abovementioned equation is nonlinear since the force vector $\bar{\mathbf{F}}$ incorporates nonlinear terms. These periodic, nonlinear, ordinary differential equations are solved to yield the blade steady response. Here, the FE in time is applied in conjunction with the Newton-Raphson method. We now discretize Eq. (6) over $N_t$ time elements around the circumference or the rotor disk (where $\psi_1 = 0, \psi_{N_t+1} = 2\pi$). Then, we consider a first order Taylor's series expansion about the steady-state value $\mathbf{y}_0 = [\mathbf{p}_0^{\mathrm{T}} \;\; \dot{\mathbf{p}}_0^{\mathrm{T}}]^{\mathrm{T}}$. This process yields the following algebraic equations.

$$\sum_{i=1}^{N_t}\int_{\psi_i}^{\psi_{i+1}} \delta\mathbf{y}_i^T\mathbf{Q}_i(\mathbf{y}_0+\triangle\mathbf{y})d\psi = \sum_{i=1}^{N_t}\int_{\psi_i}^{\psi_{i+1}} \delta\mathbf{y}_i^T[\mathbf{Q}_i(\mathbf{y}_0)+\mathbf{K}_{ti}(\mathbf{y}_0)\triangle\mathbf{y}]d\psi = 0 \tag{7}$$

$$\mathbf{K}_{ti} = \begin{bmatrix} \frac{\partial\bar{\mathbf{F}}}{\partial\mathbf{p}} - \bar{\mathbf{K}} & \frac{\partial\bar{\mathbf{F}}}{\partial\dot{\mathbf{p}}} - \bar{\mathbf{C}} \\ 0 & \bar{\mathbf{M}} \end{bmatrix}_i \tag{8}$$

Here, $\mathbf{K}_{ti}$ represents the tangential stiffness matrix for time element $i$. Furthermore, $\mathbf{Q}_i$ is the load vector for time element $i$. The modal displacement vector can be written as follows:

$$\mathbf{p}_i(\psi) = \mathbf{H}(s)\mathbf{r}_i \tag{9}$$

Here, $\mathbf{H}(s)$ are time shape functions (in terms of the element coordinates) which are fifth-order Lagrange polynomials [8] used for approximating the normal mode coordinate $\mathbf{p}$. $\mathbf{r}$ is the temporal nodal coordinate needed to describe the variation of $\mathbf{p}$ within the element. Continuity of generalized displacements is enforced between the time elements. A Lagrange–Hermite polynomials are used for interpolation inside the time element. For the numerical results, a uniform mesh with eight time elements is used.

Now, we substitute Eq. (9) and its derivative into Eq. (7). Thereafter, an iterative solution provides the blade steady response.

### 2.5 Aerodynamic loads

The air velocity in the blade-deformed plane is computed first. The blade airloads in the rotating deformed frame are then determined after applying two-dimensional strip theory. Wake model is used for the inflow and time domain unsteady aerodynamics is invoked.

### 2.6 Rotor and hub loads

The force summation method is applied to determine the steady and vibratory components of the rotating frame blade loads. In the force summation method, the blade inertia and aerodynamic forces are first integrated along the length of the blade. Then, the fixed frame hub loads are computed after summing the contributions of individual blades at the root.

### 2.7 Coupled trim

The helicopter must be trimmed for a proper assesment of the loads. The trim procedure for the helicopter mandates calcuating the pilot control angles $\mathbf{\Theta}$ which cause the six steady forces and moments acting on the helicopter to vanish. The solution of the nonlinear trim equations is performed using a Newton-Raphson procedure and many iterations are typically performed. A process known as coupled trim is conducted to solve the pilot input trim controls, blade steady response and vehicle orientation, simultaneously. This method is called coupled trim because the blade response equations and trim equations are simultaneously solved to incorporate the effect of elastic blade deflection on the rotor steady forces. Further information about the aeroelastic analysis including derivations of the blade governing equations is expounded in reference [21].

## 3 Stochastic analysis via machine learning

### 3.1 General computational framework

A physical system governed with the help of a set of equations (for example, differential equations), the general input-output functional form of the model can be expressed as

$$\mathbf{y} = \mathbf{M}(\mathbf{x}) \tag{10}$$

where $\mathbf{x} \in \mathbb{R}^M$ is a vector of input parameters of the model, representing the geometrical details, the material model and the loading. $\mathbf{y} \in \mathbb{R}^Q$ is the vector of response quantities of interest such as,

- The displacement response or its related components,
- Natural frequency, modal contribution factors and other response components in the eigen-space,
- Strain and stress component tensor at specified locations,
- Plastic strain and other internal damage measuring metrics,
- Spatial and temporal evolution of the above quantities.

Here, our motive is to set up a generalized non-intrusive framework for UQ, in which the computational model $\mathbf{M}$ can be construed as a black box, i.e., the model configuration settings cannot be edited by the user at any point and will only yield unique response values for each combination of the input vector. Also, $\mathbf{M}$ is deterministic in nature as repeating the analysis with the same set of input parameters more than once will lead to the same exact value of the output response quantity. The stochastic input parameters can be modelled by random realizations of the vector $\mathbf{x} \in \mathbb{R}^M$ in accordance to the particular probability density function $f_x(\mathbf{x})$. The conventional techniques involve the use of statistical inference based approaches, such as maximum likelihood estimate and criteria like, Akaike and Bayesian information for selecting the best fit distribution [45, 51]. Alternative approaches include

Bayesian statistics which can supplement the model prediction by utilizing measurement data in conjunction with the system physics and the maximum entropy approach for cases where there is scarce or no data.

### 3.2 Gaussian process modelling

The Gaussian process (GP) is a surrogate modelling technique which fits probability distributions over functions. GP is a spatial interpolation technique originally developed for geostatistics [28].

The functional form is expressed below by considering an independent variable $\mathbf{x} \in \mathbb{R}^d$ and function $g(\mathbf{x})$ such that $g : \mathbb{R}^d \rightarrow \mathbb{R}$, a GP over $g(\mathbf{x})$ with mean $\mu(\mathbf{x})$ and covariance function $\kappa(\mathbf{x}, \mathbf{x}'; \Theta)$ can be defined as

$$\begin{aligned} &g(\mathbf{x}) \sim GP(\mu(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}'; \Theta)), \\ &\mu(\mathbf{x}) = \mathbb{E}[g(\mathbf{x})] \\ &\kappa(\mathbf{x}, \mathbf{x}'; \Theta) = \mathbb{E}[(g(\mathbf{x}) - \mu(\mathbf{x}))(g(\mathbf{x}') - \mu(\mathbf{x}'))] \end{aligned} \tag{11}$$

where $\Theta$ represents the hyperparameters of the covariance function $\kappa$. The covariance function $\kappa$ models any prior knowledge about $g(\mathbf{x})$ and can cope with the approximation of arbitrary complex functions. In a way, the covariance function brings in interdependencies between the function value corresponding to different inputs. The squared exponential (Gaussian) covariance function illustrated in Eq. (12) is used here.

$$\kappa(\mathbf{x}, \mathbf{x}') = \sigma_g^2 \exp\left[-\sum_{i=1}^{d} \frac{(x(i) - x'(i))^2}{2r_i^2}\right] \tag{12}$$

where $\{\sigma_g, r_1, \ldots, r_d\} = \Theta$ are the hyperparameters of the covariance function.

One perspective of viewing GP is the function space mapping describing the input-output relationship [45]. As opposed to conventional modelling techniques which employ fitting a parameterized mathematical form to map the input-output functional space, a GP does not assume any explicit form, instead holds a prior belief (in the form of the mean and covariance function) onto the space of model (response) functions. Thus, GPs can be classified as a 'non-parametric' model as the number of parameters in the model are governed by the number of available data points.

Universal Kriging (a general form of GP) is employed here [33]. This constitutes a second-order polynomial trend function and GP as shown below

$$\mathbf{Y}(\mathbf{x}) = \sum_{j=1}^{p} \beta_j \mathbf{f}_j(\mathbf{x}) + \mathbf{Z}(\mathbf{x}) \tag{13}$$

where $\boldsymbol{\beta} = \{\beta_j, j = 1, \ldots, p\}$ is the vector of unknown coefficients and $\mathbf{F} = \{\mathbf{f}_j, j = 1, \ldots, p\}$ is the matrix of polynomial basis functions. $\mathbf{Z}(\mathbf{x})$ is the GP with zero mean and autovariance $\mathrm{cov}[\mathbf{Z}(\mathbf{x}), \mathbf{Z}(\mathbf{x}')] = \sigma^2 \mathbf{R}(\mathbf{x}, \mathbf{x}')$, where $\sigma^2$ is the process variance and $\mathbf{R}(\mathbf{x}, \mathbf{x}')$ is the autocorrelation function.

The parameters $\boldsymbol{\beta}$ and $\sigma^2$ can be estimated by the maximum likelihood estimate (MLE) defined by the following optimization problem under the assumption that the noise $\mathbf{Z} = \mathbf{Y} - \mathbf{F}\boldsymbol{\beta}$ is a correlated Gaussian vector

$$\begin{aligned} (\hat{\boldsymbol{\beta}}, \hat{\sigma^2}) = \arg\max_{\boldsymbol{\beta}, \sigma^2} \mathbf{L}(\boldsymbol{\beta}, \sigma^2 | \mathbf{y}) = \frac{1}{((2\pi\sigma^2)^n \det \mathbf{R})^2} \\ \exp\left[-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{F}\boldsymbol{\beta})^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{F}\boldsymbol{\beta})\right] \end{aligned} \tag{14}$$

Upon solving Eq. (14), the estimates $(\hat{\boldsymbol{\beta}}, \hat{\sigma^2})$ can be obtained as

$$\hat{\boldsymbol{\beta}} = (\mathbf{F}^T \mathbf{R}^{-1} \mathbf{F})^{-1} \mathbf{F}^T \mathbf{R}^{-1} \mathbf{y} \tag{15}$$

$$\hat{\sigma^2} = \frac{1}{n}(\mathbf{y} - \mathbf{F}\boldsymbol{\beta})^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{F}\boldsymbol{\beta}) \tag{16}$$

where $\mathbf{y}$ represents the model response such that $\mathbf{y} = \{y_1, \ldots, y_n\}^T$.

The prediction mean and variance by GP can be obtained as

$$\mu_{\hat{Y}}(\mathbf{x}) = \mathbf{F}^T \hat{\boldsymbol{\beta}} + \mathbf{r}^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{F}\hat{\boldsymbol{\beta}}) \tag{17}$$

$$\sigma^2_{\hat{Y}}(\mathbf{x}) = \hat{\sigma^2}[1 - \mathbf{r}^T \mathbf{R}^{-1} \mathbf{r} + \mathbf{u}^T (\mathbf{F}^T \mathbf{R}^{-1} \mathbf{F})^{-1} \mathbf{u}] \tag{18}$$

where $\mathbf{u} = \mathbf{F}^T \mathbf{R}^{-1} \mathbf{r} - \mathbf{R}$ and $\mathbf{r}$ are the autocorrelation between unknown point $\mathbf{x}$ and each point of the observed dataset.

Some unique features of the above formulation are: (i) The prediction is exact at the training points and the associated variance is zero. (ii) It is asymptotically zero which means as the size of the observed dataset increases, the overall variance of the process decreases. (iii) The prediction at a given point is considered as a realization of a Gaussian random variable. Thus, it is possible to derive confidence bounds on the prediction. Other adaptive versions of GP can be found in [13, 14, 38].

### 3.3 Convolution neural networks (CNN)

The convolutional neural network is a deep learning feed-forward neural network that has revolutionised computer vision in recent years. It is typically adopted in image and video recognition [56]. The main advantage of the CNN is its inherent ability to perform feature engineering from data automatically without the need for a manual feature engineering procedure, which is a long-standing bottle-neck in traditional machine learning [23]. Figure 2 shows the structure of the typical CNN. Within the model, the input layer develops a feature graph from the input data, which

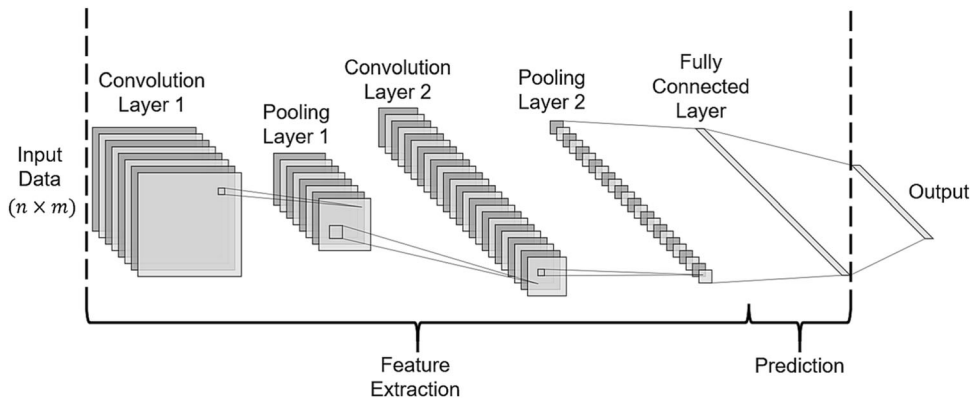


**Fig. 2** Structure of a typical CNN

corresponds to the convolution kernel. This kernel uses a set of weights to develop this feature graph. The link between the input and convolution layer is established by a receptive field, which is a square matrix of weights having sizes smaller than the input vector. As the receptive field strides along the input, it executes the convolution operation, described using the equations

$$y_{ij} = \sigma\left(\sum_{r=1}^{F}\sum_{c=1}^{F} w_{rc} x_{(r+1\times S)(c+j\times S)} + b\right) \tag{19}$$

$$0 \le i \le \frac{H-F}{S}, \quad 0 \le j \le \frac{W-F}{S} \tag{20}$$

where $y_{ij}$ is the output of a node, $H$ and $W$ represent the height (vertical) and width (horizontal) dimensions of the input, respectively. $F$ represents the height and width size of the receptive field; and $S$ denotes the stride length. The term $x_{(r+1\times S)(c+j\times S)}$ refers to the input data element with coordinates $(r+1\times S)(c+j\times S)$, and $w_{rc}$ and $b$ denote the weight positioned on the receptive field and the bias, respectively. Also, $\sigma$ represents the nonlinear activation function used to extract the features from the input.

Within the convolution layer, the input size $(H \times W \times D)$ is reduced to $[\frac{H-F+2P}{S+1} \times \frac{W-F+2P}{S+1} \times K]$, where $K$ denotes the number of filters. This process progressively decreases the dimension as the convolution layer stack becomes deeper. The pooling layer performs two key functions: (i) reduce the spatial dimension of the input layer by (typically) up to 75% and (ii) control overfitting.

Within this study, a multi-task learning deep neural network architecture is adopted. According to [11], multi-task learning applies an inductive transfer learning mechanism to improve performance in neural networks. In other words, multi-task learning trains tasks in parallel, using a shared or common representation. In multi-task learning, a neural network is trained jointly for multiple tasks and has been proven to improve predictive performance in tasks, with the prerequisite being that the tasks share conceptual similarity, or are not in competition [50]. Figure 3 represents the multi-task learning architecture adopted in this research study. As can be seen, the spine of the network is a fully connected network (FCN) feature extracted convolutional block (the shared convolutional block). Consequently, let us represent the shared convolutional block as:

$$f(x;\theta) \tag{21}$$

where $x \in \mathcal{X}$ and $\theta$ represent the parameters of the function $f$. Therefore, for each output, $y$, we define an output function $g_y(f(x);\theta_y)$, where $\theta_y$ are the parameters from the output-specific layer and $y \in \mathcal{Y}$, where $\mathcal{Y}$ denotes the set of outputs. For this study, the function $f$ of the shared convolutional block is approximated using a 1-dimensional fully connected convolutional network (FCN), as depicted in Fig. 2. Consequently, the CNN and FCN networks are gold standard state-of-the-art deep neural networks, typically adopted for computer vision and image recognition tasks. However, the FCN and CNN networks have been used in non-image classification by acting as a feature extractor.

Similarly, the model was trained in a greedy, layer-wise manner, which involves successively adding a new hidden layer to the model and refitting, allowing the newly added model to learn the inputs from the existing hidden layer, while keeping the weights for the existing hidden layers fixed. This aids to reduce/eliminate the vanishing gradient problem encountered in neural networks. Consequently, each layer was trained sequentially starting from the bottom layer (input), with each subsequent layer learning a higher-level representation of the layer below [4, 26].

### 3.4 Multi-layer perceptron (MLP)

Artificial neural networks (ANN) are considered as complex predictive models, due to their ability to handle multi-dimensional data, nonlinearity, and adept learning ability and generalisation [23]. The basic framework of a neural network incorporates four atomic elements, namely:

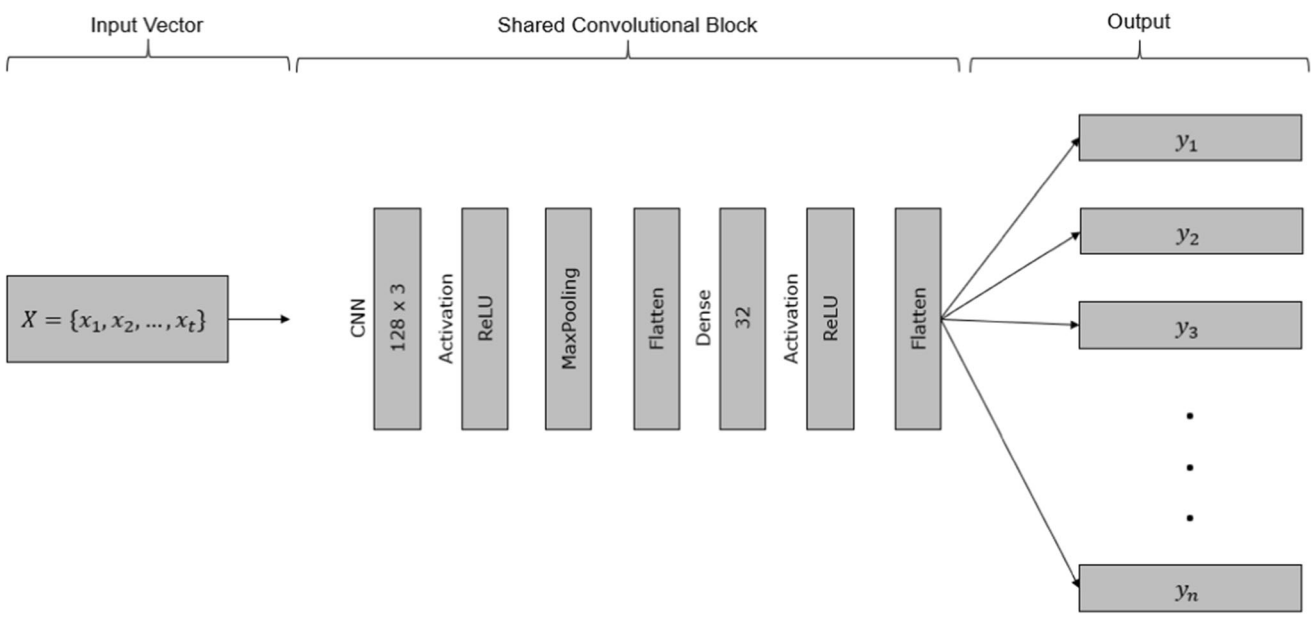


**Fig. 3** Multi-task deep neural network adopted in this study

(i) nodes, (ii) connections/weights, (iii) layers, and (iv) activation function. In the MLP, the neurons represent the building blocks. These neurons, which are simple processing units, each have weights that return weighted signals and an output signal, which is achieved using an activation function. The MLP reduces error by optimisation algorithms or functions, such as backpropagation [10, 25, 47].

In an MLP, the set of nodes are connected together by weighted connections, which can be analogous to the coefficients in a regression equation. These weighted connections represent the connecting interactions. The optimal weights of each connection between a set of layers are calculated during each backward pass of a training dataset, which is also used for weight optimisation using the derivatives obtained from the input and predicted values of the training data [24]. The layers represent the network topology, representing neuron interconnections. Within the network, the transfer function or activation function represents the transfer function or state of each neuron. The basic process in a single neuron is presented in Fig. 4. In the MLP, an external input vector is fed into the model during training. In the case of binary classification problems, during the training, the output is clamped to either 0 or 1, via the sigmoid activation function. For this present study, given that the nature of the study was regression-based, real-valued forecasting was performed using real-valued loss functions, such as mean squared error (MSE). A particular variation of neural networks is the feed-forward neural network. This is widely used in modelling many complex tasks, with the generic architecture depicted in Fig. 5. As the figure shows, the elementary model structure comprises three layers, namely the input, hidden, and output layers, respectively. In feed-forward neural networks (FFNN), each individual neuron is interconnected to the output of each unit within the next layer.

Consequently, it has been proven that an MLP, trained to minimise a loss or cost function between an input and output target variable using sufficient data, can accurately produce an estimate of the posterior probability of the output classes based on the discriminative conditioning of the input vector, which is the applied approach in this study.

### 3.5 Random forest

The Random Forest algorithm is an ensemble learning algorithm—these are algorithms that obtain their final results as aggregates of the individual forecasts of the many generated classifiers. In other words, the random forest comprises a collection of $\mathcal{T}$ tree-structured classifiers $\{T_1(X, \theta_1), T_2(X, \theta_2), \ldots, T_{\mathcal{T}}(X, \theta_{\mathcal{T}})\}$, where $X = \{x_1, x_2, \ldots, x_p\}$ is a $p$-dimensional independent and identically distributed (i.i.d) random vector of input features, and each $\theta_i \in \mathbb{R}$ represents the parameters for each individual classifier, which casts its vote for the most popular class at the input vector $X$. The output of the ensembles contains $\mathcal{T}$ outputs, $\{\hat{Y}_1 = T_1(X), \hat{Y}_2 = T_2(X), \ldots, \hat{Y}_{\mathcal{T}} = T_{\mathcal{T}}(X)\}$, where $\hat{Y}_t = 1, \ldots, \mathcal{T}$ is the predicted class from the $t$-th tree. The final output is an aggregate of all the predicted classes, which is the class with the majority vote.

The training procedure for the random forest algorithm is as follows. Consider a dataset comprising $n$ samples,

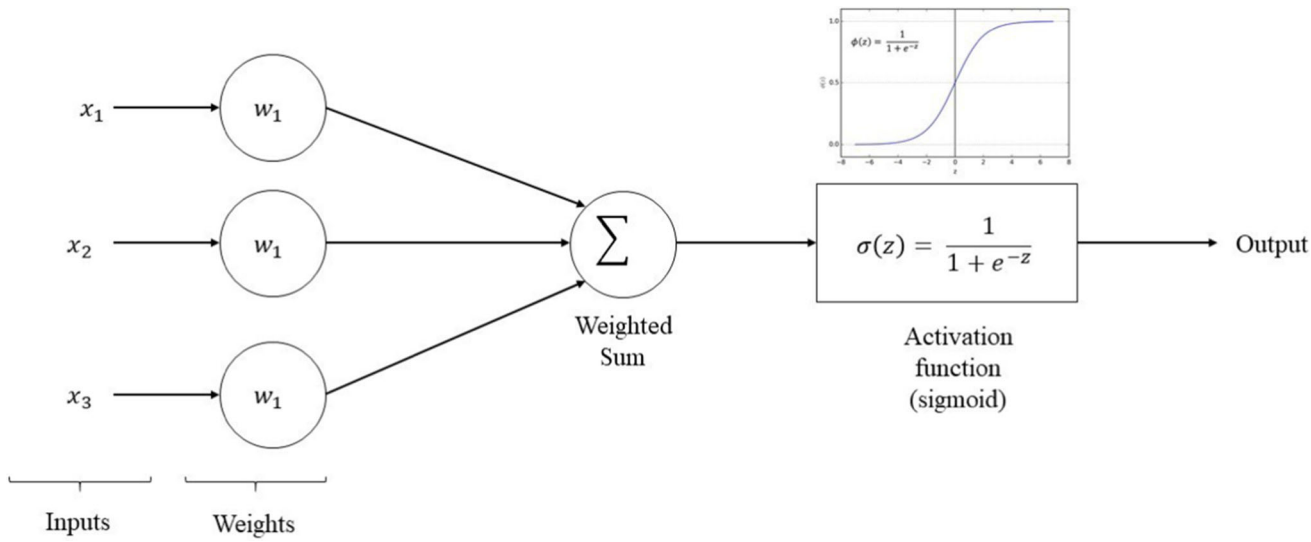


**Fig. 4** Schematic representation of the basic process in a single neuron

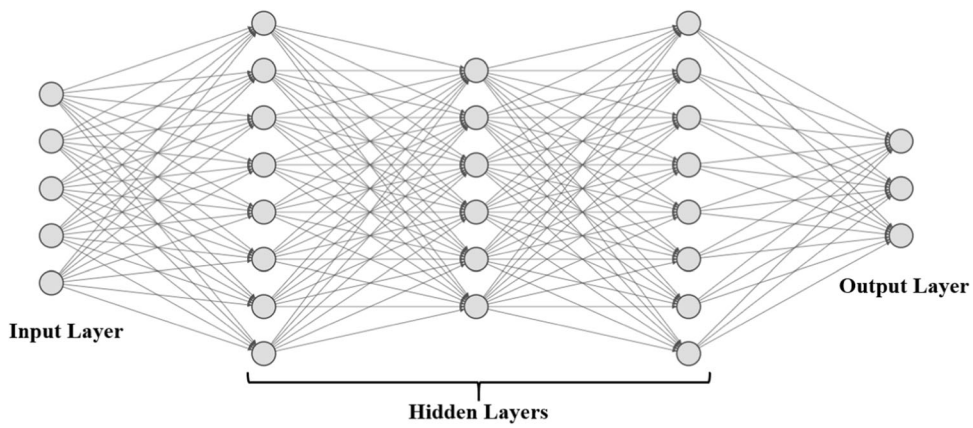


**Fig. 5** Multi-task deep neural network adopted in this study

$\mathcal{D} = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$, where $X_i$, $i = 1, 2, \ldots, n$ is a vector of input features and $Y_i$ corresponds to the class label (i.e., True or False in binary classification). Training a random forest on this dataset is as follows.

1. Draw a randomly sampled bootstrap observation (with replacement) from the $n$ observations in the training data.
2. From this bootstrap, grow a tree using the rules: select the best split from a (randomly selected) subset of $m$ features. In other words, keep $m$ as a tuning parameter for the algorithm. Grow the tree until no further split is possible, and the tree is also not pruned back.
3. Repeat the preceding steps until $\mathcal{T}$ trees are grown. When $m = n$, then the best split at each node is selected from all the features.

For this study, given that the focus is on the inference of a numerical outcome data $Y$, the random forest regressor function from the *scikit-learn*[1] package in Python was adopted instead of the classifier. The assumption that the input training data are independently drawn from the joint distribution of $(X, Y)$ and is made up of $n(p+1)$ tuples $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. In the regressor, the random forest prediction function is an unweighted average over the collection of individual learners $h(x) = (1/K)\sum_{k=1}^{K} h(\mathbf{x}; \theta_k)$.

### 3.6 Support vector machine

Classical learning algorithms are trained by minimising the error on the training dataset and this process is called

[1] RandomForest Regressor documentation can be found online at: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

empirical risk minimisation (ERM). Many machine learning algorithms learn using ERM, such as neural networks and regression-based algorithms. However, the support vector machine is based on the structural risk minimisation (SRM) principle, a statistically relevant method [46]. Some studies have proven that this method results in improved generalisation performance, given that SRM is obtained by reducing the upper bound of the generalisation error [9, 16, 49]. The support vector algorithm was developed by Russian statisticians Vapnik and Lerner [52].

To describe the inner working of the SVM, consider input data $X = \{x_i, x_2, \ldots, x_n\}$, where $n$ represents the number of samples having two distinct classes (i.e., True and False). Assume each class associated to label $y_i = 1$ for true and $y_i = 0$ for false. For linear input data, we define a hyperplane $f(X) = 0$ that separates the given data. We define a linear function $f$ of the form:

$$f(X) = W^T X + b = \sum_{j=1}^{m} w_j x_j + b = 0 \tag{22}$$

where $W \in \mathbb{R}^{n\times 1}$, and $b$ is a scalar. Together, the vector, $W$, and $b$ can be used to define the position of the hyperplane. The output of the model uses $f(X)$ to create a hyperplane that classifies the input data to either class (i.e., True or False). It is important to note that, for an SVM, the satisfying conditions for the hyperplane can be presented as

$$y_i f(x_i) = y_i(W^T x_i + b) \geq 1, i = \{1, 2, \ldots, n\} \tag{23}$$

For nonlinear classification tasks, the kernel-based SVM can be adopted. In this case, the data to be classified are mapped to a high-dimensional feature space where linear separation using a hyper-plane is possible. Consider a nonlinear vector, $\Phi(X) = (\phi_1(X), \ldots, \phi_l(X))$, which can be used to map the $m$-dimensional input vector $X$ to an $l$-dimensional feature space. The linear decision function, therefore, used to make this transformation can be given as

$$f(X) = sign\left(\sum_{i,j=1}^{n} \alpha_i y_i (\Phi(X_j) + b\right) \tag{24}$$

Although using SVM for nonlinear classification by working in the high-dimensional feature space results in benefits, for instance modelling complex problems, there are drawbacks, brought about by excessive computational requirement and overfitting. The SVM described in this section is traditionally a classifier, indicating that it is mainly applied to classifcation problems. However, the problem under investigation is a regression (i.e., real-valued forecasting) problem. For this class of problems, the support vector regression (SVR) algorithm is applied instead. The SVR still adopts the same properties as the SVM, but replaces the decision boundary in the classification problem with a match between the vector and a position in the data plane [20]. Consequently, the support vectors participate to find the closest match between the data points and the actual function representing them.

## 4 Numerical study

### 4.1 Description of the parametric stochastic model

A soft-inplane hingeless rotor with four main rotor blades which is similar to the BO105 rotor is addressed in this paper [21]. The results are generated at a non-dimensional forward speed (advance ratio) of 0.3. The rotor has 4 blades, a radius ($R$) of 4.94 m, hover tip speed ($\Omega R$) of 198.12 m/s, chord equal to 8 percent of radius, rotor solidity of 0.10 and a thrust coefficient to solidity ratio of 0.07. The mass per unit length of the blade ($m_0$) is 6.46 kg/m. In this section, the influence of material and geometric uncertainties on the (a) cross-sectional stiffness (b) natural frequencies of the non-rotating and rotating blades and (c) aeroelastic response of the composite rotor blade are investigated. In doing so, the intention is to quantify the uncertainty involved in the fabrication process and its effect on the system level response.

The blade flap bending stiffness ($EI_y$), blade lag bending stiffness ($EI_z$) and blade torsion stiffness ($GJ$) are considered to be random. These input quantities are representative of simulating the manufacturing variability as they are expressed as functions (product) of material and geometric parameters. It is intuitive to realize that material properties are prone to higher degree of variation than that of geometric parameters during a fabrication process as the former stems from a micro-mechanical model and thus, approximation in homogenization and constitutive laws may lead to modelling uncertainty being propagated to the macro-scale. Additionally, the advanced manufacturing processes have developed significantly higher precision in producing exact geometrical shapes. Having said this, the complexity associated with fabricating irregular geometries may introduce errors.

Therefore, the combined effect of material and geometric parameters are studied, with the rational outcome of the former being more sensitive on the response as it suffers from a higher level of randomness. As the random parameters involve cross-sectional stiffnesses of the rotor blade, it is practical to assume that all realizations corresponding to these parameters will be positive and therefore, they are assumed to be log-normally distributed. However, the ML-based stochastic framework employed here is generalized so that it can deal with all possible probabilistic distributions.

Murugan et al. [39] conducted a systematic study of the effect of uncertainty in composite material properties on the blade stiffnesses. They found that 5 % coefficient of variation (C.O.V.) was typical of composite material microlevel properties such as Young's modulus and Poisson's ratios. A value of 10 % C.O.V. was more representative of the higher level of dispersion which can occur for helicopter blades [42]. When the microlevel properties with 10 % C.O.V. were inserted into a composite blade analysis program, the values of C.O.V. for $EI_y$, $EI_z$ and $GJ$ were obtained as 6.88, 8.93 and 8.44, respectively. To err on the side of caution, we assume a higher value of 10 % C.O.V. on the blade stiffnesses in this paper to account for material uncertainty as well as some defects which may creep into the material during service life. The description of the random parameters is given in Table 1. Note that these quantities are non-dimensionalized by $m_0\Omega^2R^4$.

The aeroelastic response quantities of interest which are studied as part of the stochastic analysis here are: (i) blade first flap rotating frequency ($\omega_1^f$), (ii) blade second flap rotating frequency ($\omega_2^f$), (iii) blade first lag bending frequency ($\omega_1^L$), (iv) blade second lag bending frequency ($\omega_2^L$), (v) blade first torsion frequency ($\omega_1^T$), (vi) power required by main rotor ($P$), (vii) vibratory longitudinal hub force ($f_x^{4\Omega}$), (viii) vibratory lateral hub force ($f_y^{4\Omega}$), (ix) vibratory vertical hub force ($f_z^{4\Omega}$), (x) vibratory rolling hub moment ($m_x^{4\Omega}$), (xi) vibratory pitching hub moment ($m_y^{4\Omega}$), (xii) vibratory yawing hub moment ($m_z^{4\Omega}$), (xiii) vibration objective function ($J$) defined in Eq. (25), and (xiv) lowest damped stability mode (Damping).

$$J = \sqrt{(f_x^{4\Omega})^2 + (f_y^{4\Omega})^2 + (f_z^{4\Omega})^2} + \sqrt{(m_x^{4\Omega})^2 + (m_y^{4\Omega})^2 + (m_z^{4\Omega})^2} \tag{25}$$

It can be observed from Eq. (25), that $J$ can be evaluated readily from the predicted force and moment terms. In spite of this, we attempt to capture the trend of $J$ separately as an individual quantity as it is commonly used as objective function in aeroelastic design optimization frameworks [35]. This is intended for assessing the accuracy of ML techniques in capturing the stochastic trend of $J$. If the approximation quality is found to be satisfactory, there may be a huge scope towards potential application of ML in aeroelastic optimization under uncertainty frameworks in future.

For training all the ML models, 100 training points had been initially generated using a Latin hypercube sampling (LHS) scheme [37]. This was implemented with the help of the "lhsdesign" built-in function of MATLAB and the "maximin" option which maximises the minimum distance between points. However, it was observed that the actual FE model of the rotor blade did not converge for 4 of those points in the experimental design. Therefore, the ML models were eventually trained on the basis of the remaining 96 realizations of stochastic input parameters and converged aeroelastic responses.

**Table 1** Description of random parameters considered for the stochastic rotor blade analysis

| Variables | Distribution | Mean | C.O.V. |
|---|---|---|---|
| $EI_y$ | Lognormal | 0.021 | 0.1 |
| $EI_z$ | Lognormal | 0.0201 | 0.1 |
| $GJ$ | Lognormal | 0.007688 | 0.1 |

## 4.2 Adaptive Gaussian process

The MATLAB-based DACE platform was employed to implement the GP model in this work [33]. The squared exponential correlation function was used to construct the GP and 10-fold cross-validation (CV) was performed to access the model accuracy. The starting point, lower bound and the upper bound of the hyperparameters in the correlation function were adopted as, [10 10 10], [0.1 0.1 0.1] and [20 20 20], respectively, for each of the response quantities.

The root mean squared error (RMSE) obtained by GP is reported in Table 2. Note that the numerical values for all

**Table 2** RMSE values obtained from performing 10-fold CV by different ML techniques

| Responses | GP | CNN | MLP | RF | SVR |
|---|---|---|---|---|---|
| $\omega_1^f$ | **0.00012** | 0.00031 | 0.03291 | 0.01675 | 0.05014 |
| $\omega_2^f$ | 0.04709 | **0.00135** | 0.02741 | 0.02047 | 0.04865 |
| $\omega_1^L$ | **0.00255** | 0.00378 | 0.05612 | 0.03221 | 0.06135 |
| $\omega_2^L$ | 0.04609 | **0.00023** | 0.01134 | 0.01362 | 0.03975 |
| $\omega_1^T$ | 0.01275 | **0.00020** | 0.03824 | 0.01342 | 0.03858 |
| $P$ | **4.97** $\times 10^{-6}$ | 0.00025 | 0.02075 | 0.02385 | 0.05747 |
| $f_x^{4\Omega}$ | **3.25** $\times 10^{-5}$ | 0.00011 | 0.09812 | 0.00676 | 0.03721 |
| $f_y^{4\Omega}$ | **5.68** $\times 10^{-5}$ | 0.00069 | 0.01764 | 0.01345 | 0.04227 |
| $f_z^{4\Omega}$ | **0.00161** | 0.00213 | 0.06002 | 0.02177 | 0.03978 |
| $m_x^{4\Omega}$ | **1.69** $\times 10^{-5}$ | 0.00028 | 0.04848 | 0.01352 | 0.03768 |
| $m_y^{4\Omega}$ | **1.37** $\times 10^{-5}$ | 0.00084 | 0.02225 | 0.02208 | 0.04206 |
| $m_z^{4\Omega}$ | 0.00109 | **0.00032** | 0.04240 | 0.01873 | 0.04410 |
| $J$ | **9.13** $\times 10^{-6}$ | 0.00023 | 0.02454 | 0.01491 | 0.04303 |
| Damping | 0.01279 | **0.00196** | 0.01726 | 0.03234 | 0.05432 |

The lowest RMSE values corresponding to each stochastic response quantity have been indicated in bold, and illustrates the best performing ML technique

the response quantities other than the natural frequencies are small in magnitude (ranging from $10^{-1}$ to $10^{-5}$) as they are non-dimensional, and therefore, a non-relative statistical error metric (without denominator) such as RMSE has been selected in this work. It can be observed from the RMSE values in Table 2, that GP achieved a decent level of accuracy in estimating most of the response quantities. Since RMSE is not a relative error metric, in order to access the level of accuracy achieved for the respective response quantity, an idea of the magnitude of that response quantity is necessary. This can be obtained by referring to the mean values of the actual or predicted response quantities corresponding to a test dataset reported in Table 4. In particular, while approximating the following four response quantities $f_z^{4\Omega}$, $m_z^{4\Omega}$, $J$, and damping, relatively high errors by GP were observed due to their nonlinear fluctuations.

As an attempt to improve the approximation of GP for the above four response quantities, an adaptive sampling strategy by using the inherent predictive variance feature of GP is proposed. This is explained as follows:

- A GP model is built on the 96 input design points and for each of the above four response quantities (evaluated at these 96 design points).
- 10,000 random realizations of input variables are generated by MCS and the above GP model is used to predict the four response quantities corresponding to the realizations. The MSE by GP is also obtained corresponding to these 10,000 points with the help of the predictive variance feature of GP.
- The MSE obtained is normalized over the mean of the respective predicted response quantity and sorted in descending order.
- It is observed for all of the four response quantities, only the first 1000 MSE values are relatively large compared to the remaining 9000 values. So, only the first 1000 MSE values are taken into consideration to focus on the highly erroneous regions.
- The realizations of the input variables corresponding to the first 1000 MSE values are stored. This is to identify the region in the input design space where the GP predictor does not capture the response trend adequately and therefore, needs additional sample points in those regions.
- With the help of these 1000 input design points identified in the above step, the input space is partitioned into 3 clusters, corresponding to each of the four response quantities. This grouping is done according to the magnitude of the error obtained while approximating the response quantities. The purpose is to provide most number of points in the first cluster which consists of the input points for which maximum error was achieved. Similarly, the second cluster is represented by fewer points than the first cluster and more points than the third cluster. The third cluster consists of the least number of points as it corresponds to the input space for which minimum error was achieved. Intuitively the size of the original cluster will vary according to the quality of the approximation of the respective response quantity. Thus, the initial size of the first cluster of $f_z^{4\Omega}$ and $m_z^{4\Omega}$ is not necessarily identical.
- Next, the number of representative points of each cluster are obtained by the k-means clustering technique, implemented using the built-in MATLAB function 'kmeans'. For each of the response quantities, the first, second and third clusters consist of 10, 6 and 4 points, respectively. Thus, the input design space of cluster 1 is represented by more points (fine partitioning) so as to capture the response trends precisely in the erroneous regions compared to the second and third clusters. These 20 additional input points generated for each of the four response quantities are shown in Fig. 6.

Responses from the actual FE model are calculated for the above 80 additionally generated input points. However, 78 responses could be obtained as $m_z^{4\Omega}$ did not converge for two points. In order to check the improvement in the GP predictions of the four response quantities, 10-fold CV is performed on (96+20) samples of $f_z^{4\Omega}$, $J$, and damping and (96+18) samples of $m_z^{4\Omega}$. The RMSE values obtained for $f_z^{4\Omega}$ improve from 0.00161 to 0.00107, $m_z^{4\Omega}$ improves from

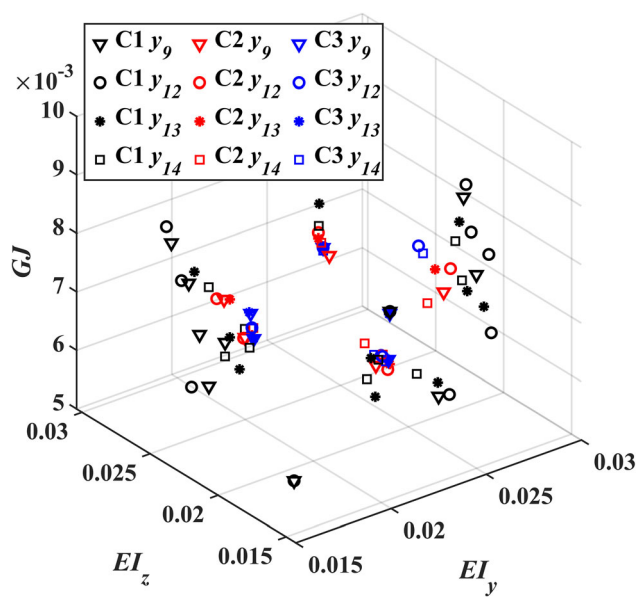


**Fig. 6** Adaptive input sample generation with predictive variance feature of GP and k-means clustering

0.00109 to 0.00071, $J$ improves from 9.13 $\times 10^{-6}$ to 8.51 $\times 10^{-6}$ and damping improves from 0.01279 to 0.00924.

For testing the performance of all the ML techniques trained on the initial 96 samples, the above 78 additionally generated points are used. This can be viewed as a type of concise MCS as all of the above 78 points resulted from applying k-means clustering on 10,000 MCS samples (as discussed previously). Moreover, these 78 points correspond to the input design space where the GP prediction was erroneous, thus, it is expected that all the ML techniques will be put to severe test upon subjecting them to this test dataset.

### 4.3 Convolution neural network: implementation details

In this study, the FCN neural network adopted is (see Fig. 2) a multi-task learning approach as depicted in Fig. 3. For its implementation, the Python-based Tensorflow software has been used [1]. In our model, the shared convolutional block had a CNN and Max pooling layer, followed by a fully connected (i.e., dense) layer and the regression layer. The CNN had 128 filters, with a kernel of length $(1 \times 2)$. A pooling layer of size $(1 \times 2)$ is applied after the CNN layer, after which a flatten layer is applied to transform the extracted features from the CNN and pooling layers to a one-dimensional vector. The fully connected (dense) layer is applied after the flatten layer to perform representation learning between the one-dimensional vector and labels. Finally, the regression layer is applied with a linear activation function to learn to make the inferences. In order to make the input data compatible with the CNN, it must be transformed to a manner that can be accepted as input. For CNNs, two-dimensional data inputs (i.e., having $n_{rows} \times n_{columns}$) must be transformed to 3-dimensional tensors, corresponding to $(n_{timesteps} \times n_{rows} \times n_{columns})$. Therefore, for this current study, each data point in the original input with dimension $(1 \times 14)$ (i.e., number of responses) is reshaped to a 3-dimensional image corresponding to $(1 \times 1 \times 14)$, treating it as a single instance of an image comprising $(1 \times rows \times columns)$. During model training, input data with three variables are used to train the model in a shared training approach, which is multi-task learning, in such a manner that the predicted responses are all learnt in a single training regime.

To optimise the parameters within a model, stochastic gradient-based optimisation algorithms are generally used. For this study, the Adam optimizer was adopted. The learning rate value was determined as $(1 \times 10^{-6})$ using a grid search mechanism. This study adopted a loss function based on the RMSE. Therefore, the RMSE was calculated on the training data to update the model parameters with each iteration (epoch).

The mini-batch stochastic gradient descent was applied using the Adam optimiser to minimise the RMSE. The performance of deep neural networks depends on predetermined hyperparameters, which are obtained using an optimization process. Unlike model parameters, which are learned using an optimization function to minimise an objective (or loss) function, hyperparameters are not learned during the model training. Many hyperparameter optimization methods exist, such as random search, grid search, and Bayesian optimization. However, for this article, we applied a grid search framework for hyperparameter optimization of all the machine learning models adopted [?].For this study, the hyperparameter optimization method can be described in the following manner: Consider a dataset $\Phi$, with an index of $n$ possible hyperparameters $h$. The grid search method simply requires the selection of a set of values for each hyperparameter $(h_1 \ldots h_k)$ that minimizes the validation loss. In other words, the grid search algorithm executes all the possible combinations of values in a 'grid' format, such that the number of trials in a grid search is $S = \prod_{n-1}^{n} |h^{(k)}|$.

The information on the trainable parameters is provided as follows. First layer is the input layer, so the input is a $1 \times 3$ tensor. In the second layer (i.e., first convolution layer), the input to the layer is the output from layer 1 and since the filter size for convolution layer 1 is $(1 \times 2)$, the number of parameters in this layer is $(((1 \times n_{input} \times filter_{size}) + bias_{parameter}) \times n_{filter})$, which is $(1 \times 3 \times 2) + 1) \times 128) = 896$. For the dense (fully-connected) layers, since each layer has 32 units, the number of trainable parameters for each layer is calculated as $(1 \times n_{inputfromCNN}) + bias_{parameter}) \times n_{units})$, which is $(((1 \times 128) + 1) \times 32)$, which is 4,128.

As it is evident from the above calculations that the number of trainable parameters are significantly high and may lead to overfitting in the model. The dropout technique was applied to control the overfitting in the model. Specifically, a dropout of 0.2 (20 %) was applied in training the deep learning model. Also, some part of the training data (10 %) was allocated for model validation, using the shuffle method (i.e., randomly shuffling the training dataset).

### 4.4 Multi layer perceptron: implementation details

In this study, we adopted an MLP, which had a shared neural network block, and one hidden layer of densely connected neurons, made of 32 units in Tensorflow [1]. The network adopted the Adam optimiser, and a learning

rate of $(1 \times 10^{-3})$. Just as with the CNN, the loss function adopted was the RMSE. The model was trained for 1,000 epochs. Similar to the deep CNN model training described in Sect. 4.3, the MLP was trained using similar parameters for the optimiser. Consequently, the model training regime was run for 300 epochs with a batch size of 16 and learning rate, $\alpha = 1 \times 10^{-3}$. The first-moment exponential decay was $\beta_1 = 0.001$, while the second-moment exponential decay was set as $\beta_2 = 0.999$.

The number of parameters in each MLP layer is calculated using the formula $(n_{units} \times n_{features})$, which is $(32 \times 2) = 64$. Note that $n_{features}$ refers to the 2 connections among the 3 inputs. For the second (fully connected) block, each layer, fully connected to a response variable has $(n_{units} + bias_{parameter})$ parameters, which is $(32 + 1) = 33$ parameters. The dropout scheme adopted for the MLP model was the same as that of the CNN model to limit the overfitting.

### 4.5 Random forest: implementation details

The random forest model in Tensorflow [1] was trained using an input dataset and 10-fold CV for model parameter tuning to ensure generalisation. For the specific model adopted in this study, we selected to train a fixed number of trees in the forest. For this, the number of trees was set as 100 and given that – as earlier stated – the random forest is an ensemble method that is trained by creating multiple decision trees, the number of trees parameter is used to specify the number of trees to be used in the process. In this study, given that the total number of features $m = 3$ is relatively small, the study adopted a bagging (bootstrap aggregation) method of training the algorithm. To train our model, we adopted the MSE to be used in measuring the quality of the split, which is equivalent to variance reduction in a feature selection regime.

### 4.6 Support vector machine: implementation details

As previously stated, the support vector regressor maps the training data into a higher-dimensional feature space, using a function and subsequently computing a hyperplane that maximises the distance between the margins of the target feature. However, the support vector regressor has many parameters that must be set for accurate parameter forecasting. These parameters, which are not optimised with model training, are referred to as hyperparameters. For this study, arriving at an optimal configuration for the hyperparameters was achieved using a grid search framework. For the SVR, the key hyperparameters include the kernel type, the kernel coefficient, the regularization parameter, and the epsilon value $\epsilon$. Consequently, the selected $\epsilon$ value for this study was set as 1.0, while the kernel function used was the radial basis function (RBF). The kernel function used was defined as $\gamma = 1/((n_{features}) \times X_{variance})$, where $n_{features}$ refers to the number of input features (i.e., 3) and $X_{variance}$ denotes the variance of these input features. For implementation, Tensorflow software was utilized [1].

Note that for this study, the multitask learning framework is only applied to the deep learning models (CNN and MLP), primarily to reduce the training time required to train a model for each output response. Given that the other shallow learning models trained relatively quickly, it was not very time consuming to loop through the individual output responses in each training cycle.

### 4.7 Results and discussion

The RMSE obtained from performing 10-fold CV by different ML techniques is presented in Table 2. The lowest RMSE values corresponding to each stochastic response quantity have been indicated in bold and thereby illustrating the best performing ML technique. From the results obtained in Table 2, it can be observed that out of all the ML techniques, GP and CNN are the most accurate. The results obtained by all the ML techniques on the test dataset are presented in terms of boxplots in Fig. 7 and RMSE values in Table 3. Figure 7 and Table 3 reveal that in addition to GP and CNN, MLP also achieves a satisfactory level of accuracy. The response statistics (mean and standard deviation) of the stochastic response quantities are reported in Table 4.

It can be observed from Table 4 that the standard deviation is high for the first torsion frequency, the second flap frequency and the second lag frequency. The first lag and flap frequencies show a low effect of the elastic stiffness uncertainty due to their strong dependence on the rotation speed. Vibration levels can increase substantially when the rotor frequencies approach multiples of the main rotor speed. Regions for the safe operation of the main rotor in terms of RPM are selected by carefully avoiding the reasons where rotating frequencies approach the multiples of the rotating speeds. The results in this paper show that an uncertainty analysis must be conducted to ensure that material uncertainty does not cause frequency shifts which can result in high vibration levels.

As can be expected, the effect of uncertainty of the stiffnesses on the rotor power is much less, as this is a higher order effect. The six vibratory loads consists of three vibratory forces and moments acting on the rotor hub. Vibratory hub loads transmitted by the main rotor to the fuselage is the main cause of vibration. The three vibratory forces are the longitudinal, lateral and vertical forces and

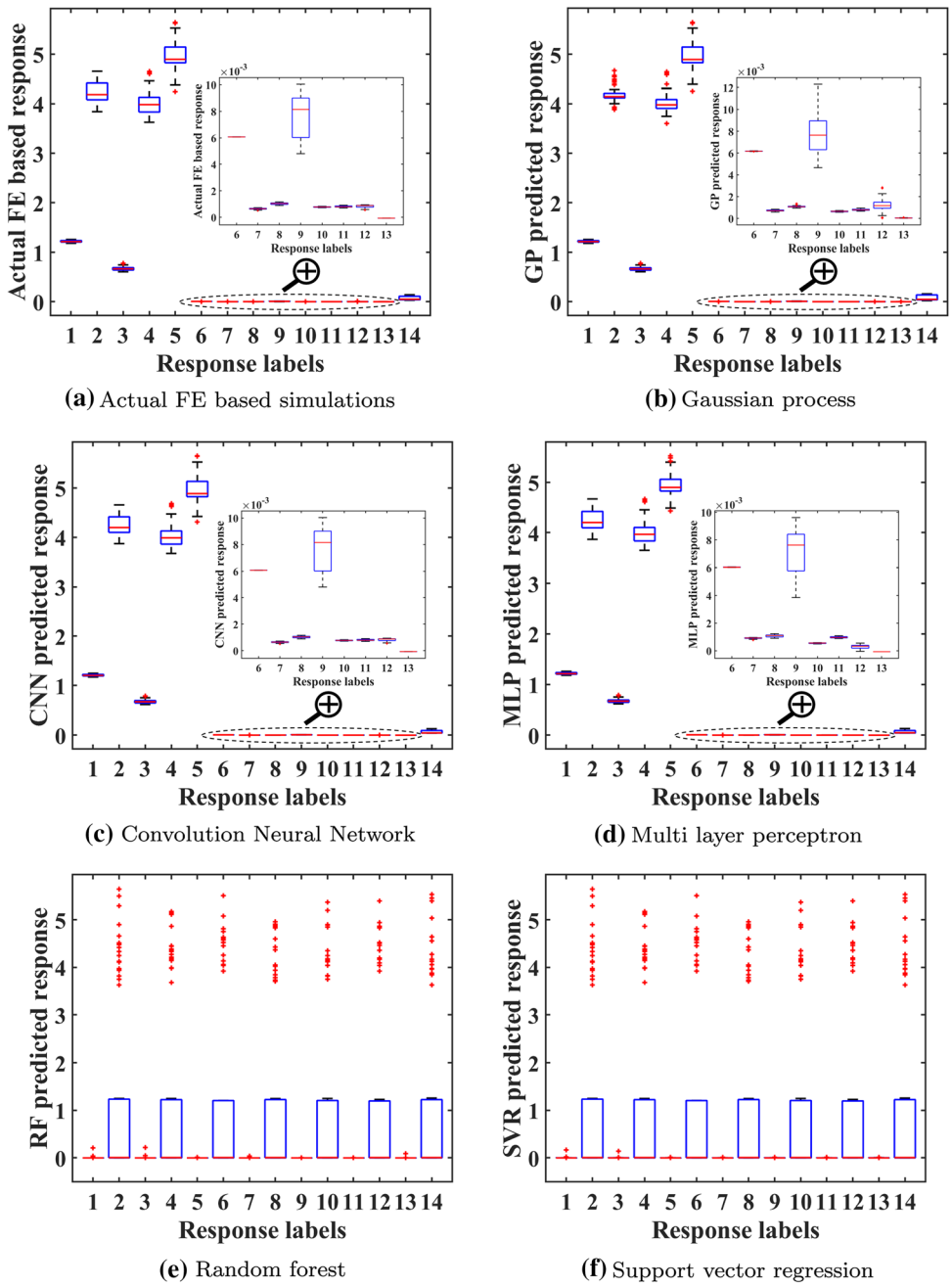


**Fig. 7** Boxplots of the stochastic response quantities corresponding to the test dataset by (**a**) actual FE-based simulations (**b**) GP (**c**) CNN (**d**) MLP (**e**) RF (**f**) SVR

are indicated by subscripts $x$, $y$ and $z$, respectively. The three vibratory moments are the rolling, pitching and yawing moment and are indicated by subscripts $x$, $y$ and $z$, respectively. The substantial effect of uncertainty can be clearly observed on the six vibratory hub loads. In particular, a high impact of uncertainty relative to the mean is seen in the yawing hub moment. The cumulative effect of uncertainty on the helicopter is shown in $J$, and again it can be seen to be quite substantial relative to the mean. Considerable effect of uncertainty is also shown in the damping. Damping in the modes for the periodic system is indicative of the possibility of the aeroelastic instability known as flutter. Typically, flutter occurs when damping becomes negative and this is a self excited oscillation which can cause the amplitude of motion of the rotor to increase inexorably until failure. While lag dampers are often used to alleviate damping, the uncertainty results show that sufficient factor of safety must be used in lag damper design to alleviate the effect of perturbation in the damping simulation results due to uncertainty in the material properties.

These results indicate that a robust and reliability design optimization approach is needed for helicopter optimization. The GP, CNN and MLP methods are shown in this

**Table 3** RMSE values obtained from approximation of the test dataset by different ML techniques

| Responses | GP | CNN | MLP | RF | SVR |
|---|---|---|---|---|---|
| $\omega_1^f$ | **0.001124** | 0.004763 | 0.002725 | 1.212949 | 1.212918 |
| $\omega_2^f$ | 0.150462 | **0.099628** | 0.101499 | 3.625389 | 3.625389 |
| $\omega_1^L$ | 0.002497 | **0.001488** | 0.003031 | 0.669514 | 0.669522 |
| $\omega_2^L$ | 0.097519 | **0.06667** | 0.067394 | 3.417177 | 3.417177 |
| $\omega_1^T$ | **0.05913** | 0.061847 | 0.08274 | 4.972982 | 4.97219 |
| $P$ | **7.11** $\times 10^{-6}$ | 0.000155 | 0.000135 | 2.063416 | 2.063416 |
| $f_x^{4\Omega}$ | **4.33** $\times 10^{-5}$ | $8.89\times 10^{-5}$ | 0.000194 | 0.004356 | 0.001848 |
| $f_y^{4\Omega}$ | **6.76** $\times 10^{-5}$ | $8.57\times 10^{-5}$ | 0.000101 | 1.999475 | 1.999475 |
| $f_z^{4\Omega}$ | 0.00151 | 0.001518 | **0.001131** | 0.007447 | 0.006816 |
| $m_x^{4\Omega}$ | **0.000674** | 0.000692 | 0.000692 | 2.034228 | 2.034228 |
| $m_y^{4\Omega}$ | **2.43** $\times 10^{-5}$ | $2.75\times 10^{-5}$ | 0.000172 | 0.001019 | 0.002021 |
| $m_z^{4\Omega}$ | **0.001781** | 0.00203 | 0.002111 | 2.077397 | 2.077397 |
| $J$ | **1.48** $\times 10^{-5}$ | $6.34\times 10^{-5}$ | 0.000103 | 0.010228 | 0.002553 |
| Damping | 0.015089 | **0.00937** | 0.009964 | 2.054043 | 2.054043 |

The lowest RMSE values corresponding to each stochastic response quantity have been indicated in bold, and illustrates the best performing ML technique

paper to be the most suitable for performing uncertainty quantification for such problems. Note that typically, vibration is minimised using the objective function $J$ and constraints are imposed on the blade rotating frequencies and damping. The damping should remain positive and the frequencies should be kept away from multiples of the main rotor speed. Uncertainty can cause a deterministic design to become infeasible. From a practical perspective, uncertainty quantification allows a systematic approach to determine margins of safety which can be used in design for frequencies, vibratory hub loads and aeroelastic damping. The use of uncertainty quantification also prevents the need for overly conservative designs based on high values of factor of safety which can lead to excess weight and the resulting deleterious consequences for a flight vehicle structure.

## 5 Summary and conclusions

The novelty of the work lies in the application of advanced data-driven learning techniques, such as convolution neural networks and multi-layer perceptron, random forests, support vector machines and adaptive Gaussian process and utilizing their multi-layered structure for capturing the nonlinear response trends to develop an efficient grey-box physics-informed ML framework for stochastic rotor analysis. Specifically, this work improves upon the accuracy aspect by metamodelling the nonlinear stochastic rotor response trends by entailing limited expensive-to-generate physics-based simulations from detailed FE models. Thus, the work is of practical significance as (i) it accounts for manufacturing uncertainties, (ii) accurately quantifies their effects on nonlinear response of rotor blade and (iii) makes the otherwise computationally prohibitive simulations viable by the use of ML.

A comparative assessment of advanced deep and shallow supervised learning techniques is presented. These data-driven techniques have been trained to learn from the stochastic aeroelastic response trends and build corresponding physics-based meta-models of the system, thereby eliminating the need to perform high-fidelity simulations on the actual FE model. For simulating the manufacturing variability, the combined effect of material and geometric randomness have been taken into account.

Important findings from the results obtained in this study include:

- In general, high sensitivity of the rotor aeroelastic output responses to the input elastic stiffness uncertainty reveals that considering manufacturing variability in analyzing helicopter rotors is pivotal to simulate their actual behaviour.
- To be specific, few response parameters like the first torsion frequency, vibratory hub loads and damping have a substantial effect due to the input perturbations. The highest sensitivity has been observed in the yawing hub moment. This suggests that sufficient factor of safety should be considered in the rotor design to (i) prevent frequency shifts which can result in high vibration levels and, (ii) avoid the occurrence of the aeroelastic instability condition known as flutter and

**Table 4** Response statistics corresponding to the test dataset by different ML techniques. The response statistics are reported as mean (+/− standard deviation)

| Response | Actual | GP | CNN | MLP | RF | SVR |
|---|---|---|---|---|---|---|
| $\omega_1^f$ | 1.21591 (+/−0.02006) | 1.21602 (+/−0.01992) | 1.22051 (+/−0.01965) | 1.21836 (+/−0.01999) | 0.00338 (+/−0.02375) | 0.00331 (+/−0.01904) |
| $\omega_2^f$ | 4.23513 (+/−0.21216) | 4.19596 (+/−0.15989) | 4.23746 (+/−0.19906) | 4.25205 (+/−0.20214) | 1.10708 (+/−1.81926) | 1.10708 (+/−1.81926) |
| $\omega_1^L$ | 0.67141 (+/−0.03699) | 0.67004 (+/−0.03672) | 0.67124 (+/−0.03652) | 0.67365 (+/−0.03655) | 0.00337 (+/−0.02473) | 0.00307 (+/−0.01583) |
| $\omega_2^L$ | 4.01168 (+/−0.22134) | 4.00721 (+/−0.1824) | 4.01233 (+/−0.21672) | 4.00235 (+/−0.21757) | 1.10354 (+/−1.79866) | 1.10354 (+/−1.79866) |
| $\omega_1^T$ | 4.96449 (+/−0.29636) | 4.97205 (+/−0.29084) | 4.94809 (+/−0.28193) | 4.94683 (+/−0.23545) | 0.00023 (+/−0.0017) | 0.00103 (+/−0.00251) |
| $P$ | $0.00617(+/-1.55\times10^{-5})$ | $0.00617(+/-1.44\times10^{-5})$ | $0.00602(+/-5.46\times10^{-6})$ | $0.00604(+/-6.59\times10^{-6})$ | 1.06518 (+/−1.78239) | 1.06518(+/−1.78239) |
| $f_x^{4\Omega}$ | $0.00072(+/-7.66\times10^{-5})$ | $0.00072(+/-6.09\times10^{-5})$ | $0.00065(+/-4.90\times10^{-5})$ | $0.00091(+/-4.14\times10^{-5})$ | 0.0007 (+/−0.00438) | 0.0009 (+/−0.00184) |
| $f_y^{4\Omega}$ | $0.0011(+/-8.29\times10^{-5})$ | $0.00108(+/-8.46\times10^{-5})$ | $0.0011(+/-5.99\times10^{-5})$ | $0.00107(+/-8.24\times10^{-5})$ | 1.03456 (+/−1.72278) | 1.03456 (+/−1.72278) |
| $f_z^{4\Omega}$ | 0.00726 (+/−0.00185) | 0.00769 (+/−0.00181) | 0.00826 (+/−0.00132) | 0.00716 (+/−0.00158) | $6.31\times10^{-5}$ (+/−0.0003) | 0.00114 (+/−0.00215) |
| $m_x^{4\Omega}$ | 0.00071 (+/−0.00068) | 0.00064 ($+/-4.01\times10^{-5}$) | 0.00054 ($+/-2.33\times10^{-5}$) | 0.00055 ($+/-2.57\times10^{-5}$) | 1.04962 (+/−1.75424) | 1.04962 (+/−1.75424) |
| $m_y^{4\Omega}$ | 0.0008 ($+/-7\times10^{-5}$) | 0.0008 ($+/-6.67\times10^{-5}$) | 0.0008 ($+/-5.51\times10^{-5}$) | 0.00097 ($+/-5.34\times10^{-5}$) | 0.00011 (+/−0.00075) | 0.00091 (+/−0.00204) |
| $m_z^{4\Omega}$ | 0.00139 (+/−0.00184) | 0.00119 (+/−0.00049) | 0.00044 (+/−0.00015) | 0.0003 (+/−0.00014) | 1.06964 (+/−1.79301) | 1.06964 (+/−1.79301) |
| $J$ | $3.34\times10^{-5}$ ($+/-1.49\times10^{-5}$) | $3.75\times10^{-5}$ ($+/-1.81\times10^{-5}$) | $9.60\times10^{-5}$ ($+/-7.09\times10^{-6}$) | $-6.89\times10^{-5}$ ($+/-4.09\times10^{-20}$) | 0.00129 (+/−0.01022) | 0.00125 (+/−0.00226) |
| Damping | 0.06769 (+/−0.03426) | 0.07223 (+/−0.04581) | 0.0697 (+/−0.02894) | 0.06382 (+/−0.02727) | 1.07263 (+/−1.80762) | 1.07263 (+/−1.80762) |

accordingly, design lag dampers which are used to alleviate damping.

- The results achieved highlight the fact that CNN and GP are the most accurate models followed by MLP. RF and SVM mostly failed to capture the response trends, with a very few exceptions where some response quantities were decently predicted. The accuracy obtained by CNN, GP and MLP is worth acknowledging as (i) a high proportion of variation in the input parameters was considered (ii) the prediction test dataset consisted of the points from the stochastic input space where GP initially proved to be vulnerable. Additionally, an adaptive sampling strategy was devised by using the predictive variance feature of GP (i) to improve the accuracy by adding a nominal number of points to the experimental design and (ii) the additional points generated were used to create the test dataset in which the other models could be validated.

For extending this research in a future direction, it will be worth investigating the effect of uncertainties on different rotor models. This will create additional datasets based on different physical insights on the structural system. Also, this work does not account for spatially varying uncertainties, which may be prevalent for helicopter rotor models. This will require integration of random field models for the stochastic elastic stiffness parameters with the present computational framework and is a potential direction for future investigation. Since a decent level of accuracy is achieved by CNN, GP and MLP, these machine learning models can be extended for applications of optimization under uncertainty of composite rotor blades. Although the deep learning techniques may be hard to train, once the ideal model configuration is achieved, they can easily be employed to solve more expensive problems such as the optimal design of the blades. The capability of these methods in operating in high-dimensional spaces will be advantageous to GP and conventional surrogate modelling approaches which easily tend to collapse in these complex scenarios. Therefore, deep and shallow neural net driven robust or reliability based design of composite helicopter rotor blades for vibration control will be an interesting extension of this present work. One of the approaches to solve the optimization problem can be minimizing the vibration (denoted by the term $J$) with the constraints imposed on the blade rotating frequencies and damping to ensure the frequencies are kept away from the multiples of the main rotor speed and the damping remains positive.

**Acknowledgements** Chatterjee and Friswell gratefully acknowledge the support of the Engineering and Physical Sciences Research Council through the award of the Programme Grant 'Digital Twins for Improved Dynamic Design', grant number EP/R006768.

## Declarations

**Conflict of interest** The authors have no conflicts of interest to declare that are relevant to the content of this article.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

## References

1. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, others GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X (2015) TensorFlow: large-scale machine learning on heterogeneous systems. http://tensorflow.org/
2. Adamson L, Fichera S, Mottershead J (2020) Aeroelastic stability analysis using stochastic structural modifications. J Sound Vib 477:115333
3. Batrakov A, Kusyumov A, Mikhailov S, Barakos G (2018) Aerodynamic optimization of helicopter rear fuselage. Aerosp Sci Technol 77:704–712
4. Bengio Y, Lamblin P, Popovici D, Larochelle H et al (2007) Greedy layer-wise training of deep networks. Adv Neural Inform Process Syst 19:153
5. Beran P, Stanford B, Schrock C (2017) Uncertainty quantification in aeroelasticity. Ann Rev Fluid Mech 49:361–386
6. Bernerdini G, Piccione E, Anobile A, Serafini J, Gennaretti M (2016) Optimal design and acoustic assessment of low-vibration rotor blades. Int J Rotat Machin 2016:1–17
7. Bir G, Chopra I, Ganguli R (1992) University of Maryland advanced rotorcraft code UMARC theory manual. Tech rep, UM-AERO Report 92-02, Center for Rotorcraft Education and Research, University of Maryland, College Park
8. Borri M (1986) Helicopter rotor dynamics by finite element time approximation. Comput Math Appl 12(1):149–160
9. Boser B, Guyon I, Vapnik V (1992) A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on Computational learning theory, pp 144–152, isbn: 089791497X. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/130385.130401
10. Brink A, Najera-Flores D, Martinez C (2021) The neural network collocation method for solving partial differential equations. Neural Comput Appl 33:5591–5608
11. Caruana R (1997) Multitask learning. Mach Learn 28(1):41–75
12. Chassaing J, Nitschke C, Vincenti A, Cinnella P, Lucor D (2018) Advances in parametric and model-form uncertainty quantification in canonical aeroelastic systems. J Aerosp Lab 14:1–19

13. Chatterjee T, Chowdhury R (2018) h - p adaptive model based approximation of moment free sensitivity indices. Comput Methods Appl Mech Eng 332:572–599
14. Chatterjee T, Chakraborty S, Chowdhury R (2016) A bi-level approximation tool for the computation of FRFs in stochastic dynamic systems. Mech Syst Signal Process 70–71:484–505
15. Chatterjee T, Chakraborty S, Chowdhury R (2019) A critical review of surrogate assisted robust design optimization. Archiv Comput Methods Eng 26(1):245–274
16. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
17. Daniel M, Brewer W, Behm G, Strelzoff A, Wilson A, Wade D (2018) Deep learning evolutionary optimization for regression of rotorcraft vibrational spectra. In: IEEE/ACM Machine Learning in HPC Environments (MLHPC), Dallas, TX, USA. https://doi.org/10.1109/MLHPC.2018.8638645
18. Dempsey P, Branning J, Wade D, Bolander N (2010) Comparison of test stand and helicopter oil cooler bearing condition indicators. In: Proceedings of the American Helicopter Society 66th Annual Forum and Technology, Phoenix, AZ
19. Dey S, Mukhopadhyay T, Adhikari S (2017) Metamodel based high-fidelity stochastic analysis of composite laminates: a concise review with critical comparative assessment. Compos Struct 171:227–250
20. Drucker H, Burges C, Kaufman L, Smola A, Vapnik V et al (1997) Support vector regression machines. Adv Neural Inform Process Syst 9:155–161
21. Ganguli R (2002) Optimum design of a helicopter rotor for low vibration using aeroelastic analysis and response surface methods. J Sound Vib 258(2):327–344
22. Gennaretti M, Bernardini G, Serafini J, Romani G (2018) Rotorcraft comprehensive code assessment for blade-vortex interaction conditions. Aerosp Sci Technol 80:232–246
23. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT press, Cambridge, MA, USA. http://www.deeplearningbook.org
24. Hamdia K, Zhuang X, Rabczuk T (2020) An efficient optimization approach for designing machine learning models based on genetic algorithm. Neural Comput Appl. https://doi.org/10.1007/s00521-020-05035-x
25. Hecht-Nielsen R (1988) Applications of counterpropagation networks. Neural Networks 1(2):131–139
26. Hinton G, Osindero S, Teh Y (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7):1527–1554
27. Kontogiannis S, Demange J, Savill A, Kipouros T (2020) A comparison study of two multifidelity methods for aerodynamic optimization. Aerosp Sci Technol 97(105):592
28. Krige DG (1951) A statistical approach to some basic mine valuation problems on the witwatersrand. J Chem Metall Min Soc South Africa 52(6):119–139
29. Kumar R, Ganguli R, Omkar SN (2010) Rotorcraft parameter estimation using radial basis function neural network. Appl Math Comput 216(2):584–597
30. Kumar V, Omkar S, Ganguli R, Sampath P, Suresh S (2006) Identification of helicopter dynamics using recurrent neural networks and flight data. J Am Helicopter Soc 51(2):164–174
31. Li F, Gao L, Garg A, Shen W, Huang S (2020) A comparative study of pre-screening strategies within a surrogate-assisted multi-objective algorithm framework for computationally expensive problems. Neural Comput Appl. https://doi.org/10.1007/s00521-020-05258-y
32. Li L (2007) Structural Design of Composite Rotor Blades with Consideration of Manufacturability, Durability, and Manufacturing Uncertainties. PhD thesis, Georgia Institute of Technology. http://hdl.handle.net/1853/24757
33. Lophaven S, Nielson H, Sondergaard J (2002) DACE A MATLAB Kriging Toolbox. Computer programme, Informatics and Mathematical Modelling, Technical University of Denmark, IMM-TR-2002-12. http://www2.imm.dtu.dk/pubdb/p.php?1460
34. Lu Y, Su T, Chen R, Li P, Wang Y (2019) A method for optimizing the aerodynamic layout of a helicopter that reduces the effects of aerodynamic interaction. Aerosp Sci Technol 88:73–83
35. Mallick R, Ganguli R, Bhat M (2015) Robust design of multiple trailing edge flaps for helicopter vibration reduction: a multi-objective bat algorithm approach. Eng Optim 47(9):1243–1263
36. Martínez D, Brewer W, Strelzoff A, Wilson A, Wade D (2020) Rotorcraft virtual sensors via deep regression. J Parallel Distrib Comput 135:114–126
37. McKay M, Beckman RJ, Conover WJ (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21(2):239–245
38. Moustapha M, Bourinet JM, Guillaume B, Sudret B (2018) Comparative study of kriging and support vector regression for structural engineering applications. J Uncertain Eng Syst Part A Civ Eng 4(2):04018
39. Murugan S, Harursampath Ganguli R (2008) Material uncertainty propagation in helicopter nonlinear aeroelastic response and vibratory analysis. AIAA J 46(9):2332–2344
40. Murugan S, Chowdhury R, Adhikari S, Friswell M (2012) Helicopter aeroelastic analysis with spatially uncertain rotor blade properties. Aerosp Sci Technol 16(1):29–39
41. Muscolino G, Ricciardi G, Cacciola P (2003) Monte carlo simulation in the stochastic analysis of non-linear systems under external stationary poisson white noise input. Int J Nonlinear Mech 38:1269–1283
42. Onkar A, Yadav D (2005) Forced nonlinear vibration of laminated composite plates with random material properties. Compos Struct 70(3):334–342
43. Pettit C (2004) Uncertainty quantification in aeroelasticity: recent results and research challenges. J Aircraft 41(5):1217–1229
44. Pflumm T, Rex W, Hajek M (2019) Propagation of Material and Manufacturing Uncertainties in Composite Helicopter Rotor Blades. In: 45th European Rotorcraft Forum, Warsaw, Poland
45. Rasmussen CE, Williams CKI (2006) Gaussian processes for machine learning. The MIT Press, Cambridge, Massachusetts London, England
46. Roman I, Santana R, Mendiburu A, Lozano J (2020) In-depth analysis of svm kernel learning and its components. Neural Comput Appl. https://doi.org/10.1007/s00521-020-05419-z
47. Rumelhart D, Hinton G, Williams R (1986) Learning representations by back-propagating errors. Nature 323:533–536. https://doi.org/10.1038/323533a0
48. Saijal K, Ganguli R, Viswamurthy SR (2011) Optimization of helicopter rotor using polynomial and neural network metamodels. J Aircraft 48(2):553–566
49. Vladimir N. Vapnik (1995) The nature of statistical learning theory. Springer, New York, NY. https://doi.org/10.1007/978-1-4757-2440-0
50. Sener O, Koltun V (2018) Multi-task learning as multi-objective optimization. In: NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp 525–536. Curran Associates Inc., Red Hook, NY, United States
51. Sudret B (2012) Meta-models for structural reliability and uncertainty quantification. In: Proceedings of 5th Asian-Pacific Symposium on Stuctural Reliabilty and its Applications (APSSRA, 2012), Singapore, pp 53–76, ID: **hal-00683179**
52. Vapnik V, Lerner A (1963) Generalized portrait method for pattern recognition. Autom Remote Control 24(6):774–780
53. Vu-Bac N, Lahmer T, Zhuang X, Nguyen-Thoi T, Rabczuk T (2016) A software framework for probabilistic sensitivity

analysis for computationally expensive models. Adv Eng Softw 100:19–31
54. Wade D, Wilson A (2017) Applying machine learning-based diagnostic functions to rotorcraft safety. In: 17th Australian International Aerospace Congress: AIAC 2017, Engineers Australia, Royal Aeronautical Society, pp 663–669
55. Wade D, Vongpaseuth T, Lugos R, Ayscue J, Wilson A, Antolick L, et al (2015) Machine learning algorithms for hums improvement on rotorcraft components. In: Proceedings of the 71st Annual Forum of the American Helicopter Society, Virginia Beach, Virginia
56. Wu C, Jiang P, Ding C, Feng F, Chen T (2019) Intelligent fault diagnosis of rotating machinery based on one-dimensional convolutional neural network. Comput Ind 108:53–61

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.