Automation in Construction 138 (2022) 104233

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Metamodel-based generative design of wind turbine foundations

Qinshuo Shen [a], Faridaddin Vahdatikhaki [a,*], Hans Voordijk [a], Jeffrey van der Gucht [b], Lex van der Meer [b]

[a] Department of Construction Management and Engineering, University of Twente, Horsttoren, Drienerlolaan 5, 7522 NB, Enschede, the Netherlands
[b] WindBase, ABT, Arnhemsestraatweg 358, 6881 NK Velp, the Netherlands

ARTICLE INFO

Keywords:
Metamodels
Generative design
Simulation-based optimization
Wind turbine
Foundation design

ABSTRACT

Wind turbines play an integral role in energy transition agendas. The optimized design of wind turbine foundations is a complex and intricate task that requires iterative running of computationally-intensive and time-consuming finite element models. However, given the popularity of these structures over the past two decades, there is a wealth of data from the designs of the past projects that can be used for the data-driven modeling of these structures. Given the demonstrated accuracy and success of metamodels as an alternative approach for other computationally-intensive simulation-based problems, this study aims to develop a generative-design framework for the optimization of wind turbine foundations using a metamodel, as a complementary step to more accurate finite element modeling, to reduce the overall design time without compromising the accuracy. To this end, first, the random forest method is used to develop a multi-output metamodel for the wind turbine foundations based on a set of historical data. Then, a metaheuristic method, i.e., NSGA II, is adopted to optimize the design process based on the developed metamodel. In a case study, a wind turbine foundation was designed using the proposed framework and the accuracy of the output was assessed in terms of the ultimate bending moment. The results of the case study indicate that the proposed method provides a significant time gain (i.e., 99.93%) without compromising the accuracy (i.e., 1.75% for the percent error). Besides, the conducted study also offers designers a better understanding of the importance of each design variable and how certain design variables influence the moment-rotation behavior of the wind turbine foundation

1. Introduction

The awareness about increasing anthropogenic climate change has created a strong momentum in many countries to devise and adopt energy transition strategies to shift from fossil fuels to renewable energy [1]. A typical resource is wind energy, which is widely accepted as the fastest growing renewable energy source because of its availability, greenness, and cost-efficiency [2]. The kinetic energy from wind is captured by wind turbines, which play an integral role in harnessing wind energy. Over the past two decades, new installations and investments in wind energy boosted sharply. In 2019, approximately 15.4 GW of wind power capacity has been installed within European countries, expanding the total wind energy capacity in Europe to 205 GW, which accounts for 15% of the total electricity consumed in Europe in 2019 [3]. In order to continuously grow the share of wind energy in the new energy market, it is critical to keep expanding the wind turbine grid [4]. To realize this goal, a large number of new wind turbines need to be designed and installed. Therefore, it is essential to pay more attention to the design of wind turbines to shorten the construction process without compromising structural performance.

Wind turbines can either be onshore or offshore. Although winds are stronger and more stable in the offshore areas, onshore wind turbines still dominate the market. This is because offshore wind turbines are relatively more complex and costly to install and maintain [5]. Nearly 75% of the newly installed wind turbines in 2019 in Europe were onshore, making up 89% of the wind turbine capacity [3]. The design and construction of onshore wind turbines may strike as straightforward and repetitive. However, the unsteady aerodynamic effects caused by the reaction between the turbulent wind and blade section and the strongly coupled engineering system of wind turbines render the design process complex [6–8]. Among the structures consisted in the wind turbine system, early studies have illustrated the importance of the wind turbine foundation for the dynamic characteristics of the system [9]. Currently, static analysis is still commonly used for the design

* Corresponding author.
E-mail address: f.vahdatikhaki@utwente.nl (F. Vahdatikhaki).

https://doi.org/10.1016/j.autcon.2022.104233
Received 13 January 2021; Received in revised form 6 October 2021; Accepted 27 March 2022
Available online 1 April 2022

optimization of wind turbine foundations. This analysis focuses on finding optimal solutions using non-linear analysis, under static load cases, constraints, and objectives [7,10]. Considering the difficulties of solving non-linear problems, approximation and reduction will be made to simplify the problems using computerized numerical methods such as finite element analysis (FEA), which can provide more accurate models to approximate the actual behavior of the structure [11]. However, although the usage of other computational approaches such as parametric design streamlines the process, the design optimization still requires many iterations, analysis, and fine-tuning of the design until the best option is generated under the constraints and objective functions. Furthermore, because the simulation using FEA is time-consuming and computationally expensive, it is difficult to run a considerable number of analyses and evaluate different design alternatives.

For the past few years, due to the expansion of the wind turbine network, a wealth of data has been generated from the designs of the wind turbine foundations. Potentially, this database can be used to guide new designs towards optimality without going through a lengthy optimization process. However, this would require a metamodel that can correlate initial input to corresponding output and mathematically approximate the complex simulation model. This metamodel can be then used for rapid optimization. As a popular engineering method, metamodeling refers to the development of response surface surrogates to approximate the original simulation models using data-driven techniques, where less intensive computation, reduced noisy output behavior, and the provision of gradients can be expected [12–14].

Due to the rapid development of computer technologies, Machine Learning (ML) approaches have shown the potential to tackle the aforementioned problem. An ML model is developed through a self-learning process where the training data is used to investigate and identify patterns between a set of input and the output data [15]. Therefore, a significant time gain is expected by reducing the computational intensity caused by the current FEA simulation when the ML model can predict a near-optimal design more efficiently. It has previously been observed that the application of metamodeling techniques can greatly alleviate the extensive computational cost required for using traditional simulation models in the design of wind turbine structures [16–18]. For instance, Albanesi et al. [19] proposed a framework using an Artificial Neural Network (ANN) to create a metamodel and reduce the computational time in the optimization process of the composite laminate of wind turbine blades. Ju et al. [20] compared three types of ML algorithms, namely ANN, Support Vector Machine (SVM), and Radial Basis Function (RBF), to replace the Monte Carlo simulation method used for the uncertainty analysis of turbomachinery. Also, Kang et al. [21] developed a SVM-based simulation method for the structural reliability analysis of offshore wind turbine foundations. However, the application of ML-based metamodeling techniques has so far been limited to the superstructure of offshore wind turbines. Accordingly, the application of metamodeling for the design of onshore wind turbine foundations, to the best of authors' knowledge, has not been fully investigated in the literature. Therefore, the extent to which ML-based metamodeling techniques can streamline the design process, and thereby design optimization, of onshore wind turbine foundations is still not fully known.

Although ML can be of significant help in the design process of the wind turbine foundations, it should be highlighted that given the black-box nature of ML models and also the safety, liability, and legal issues that surround the design of assets, ML models cannot be used as a substitute for the traditional code-based design. It is important that a human designer perform a thorough compliance check of the final design with respect to structural mechanics, safety and cost. Nonetheless, ML models can be used to shorten the iterative design process that aims towards optimal design considerably by quickly directing the designer to the area of optimal design. Theoretically, designers can use ML models to find a much stronger starting point for their design iteration.

On this premise, this study aims to develop a generative design framework for the optimization of wind turbine foundations using a multi-output ML-based metamodel, as a complementary step to the more accurate finite element modeling, in order to reduce the overall iterative design time. Generative design refers to the integration of a simulation/assessment technique (e.g., a meta-model) with an optimization method to effectively explore the wide solution space and identify the optimum design alternatives.

The following contents are organized as follows. In Section 2, a literature review will be presented to provide insights into the theoretical background of this research. The third section of this article is concerned with the proposed method of metamodel development, while Section 4 demonstrates a case study and corresponding results, which can be used to validate the proposed method. Besides, Section 5 provides the discussion according to the results obtained from the conducted case study. The conclusion and future work are given in Section 6.

2. Literature review

2.1. Wind turbine foundation design optimization methods

As the supporting structure, the wind turbine foundation provides both stability and stiffness to maintain the corresponding structural requirement. Because of the interaction between the wind force and rotor blades, a large moment and lateral loads will be induced and transferred to the foundation, creating a strong tendency for the wind turbine to overturn [8]. The design of the wind turbine foundation is, therefore, highly sensitive to this aero-elastic effect, leading to extensive nonlinearities. Besides, to have optimized designs for wind turbine foundations, intending to reduce costs while safeguarding structural stability, many variables and constraints need to be considered, including the foundation geometry, the reinforcement configuration, load factors, and the project-specific soil condition [7]. However, the sheer number of pertinent parameters and variables significantly render the optimization process a demanding task.

There is much research in the area of the design optimization of wind turbine structures, particularly in computer-aided design optimization. Previous studies on the design optimization of wind turbine structures have formulated the design optimization problem mathematically to enable the use of computer-aided, automated, and algorithmic approaches [21]. Among these approaches, the static analysis is most widely used in the design optimization of wind turbine structures [7,11]. Typically, the objective of conducting this design optimization method is minimizing the material usage of the structure without compromising its structural performance. In order to solve the optimization problem under static loads, the non-linear behavior of the structure must be analyzed, thus arising the necessity of using numerical analysis methods such as FEA. Generally, the optimization process under static loads is iterative, because a number of convex approximations are required to be solved [7].

2.2. Metamodel-based design optimization

FEA models are not very suitable for the gradient assessment of the wind turbines because the structural response is too sensitive to changes in the geometry of the structure. Therefore several researchers are proposing the use of the simulation-based design optimization methods, including metamodel-based techniques, heuristic methods, and stochastic search [22,23].

Metamodel-based (or surrogate-model-based) techniques shows a great potential. Fig. 1 represents how the metamodel interacts with the original simulation model within the optimization process, by developing cheaper-to-run surrogates to fully or partly replace the original simulation model process using mathematical functions [14]. The main objective of adopting the metamodeling technique in design optimization is to solve complex optimization problems under the computational budget limitations. Because these techniques use the analytical

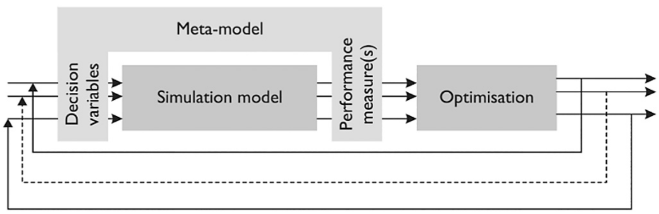

Fig. 1. The general meta-model-based design optimization framework [14].

approach to approximate the solution of the objective function rather than estimating the derivatives, they can explore the optimization space much faster [14,24].

Great attempts have been made to apply metamodeling to the Architecture, Engineering, and Construction (AEC) industry over the past few decades to solve engineering problems. Moayedi et al. [25] presented a neural-metaheuristic-based metamodel to predict the soil shear strength, which shows the efficiency of modeling the non-linearities of the soil behavior and influential soil parameters. Also, neural-network-based metamodels were deployed to predict the foundation capacity [26–28]. In addition, numerous studies also indicated how the metamodeling techniques can solve other geotechnical engineering problems with higher computational efficiency and better performance, including the subsurface exploration, slope stability assessment, and landslides assessment [29–31]. Bakalis et al. [32] demonstrated a low-fidelity 3D surrogate model for the seismic performance assessment of liquid storage tanks, which can enable the fast static and dynamic assessment by streamlining the nonlinear analysis. Mardfekri [33] proposed a probabilistic metamodel to examine the seismic fragility of wind turbine structure models faster than the conventional methods. Guo et al. [34] proposed a meta learning-based façade defects classification framework using an image dataset and the concept of deep learning. Chen et al. [35] presented a metamodel to calibrate the performance of the building energy model. Furthermore, Zheng et al. [36] proposed a machine-learning-based topological design approach to expedite the exploration process for shell structures.

Overall, above literature provides substantial evidence that the utilization of metamodeling techniques can greatly reduce the computational cost in the design processes of complex civil structures, such as wind turbines [19,20,37]. However, the application of these metamodeling techniques have been limited to the superstructure of wind turbines and, thus, not for the foundation design. Thus, there is little insight into how the design process of onshore wind turbine foundations can benefit from these techniques.

2.3. Machine learning methods

One of the most effective and widely applied approaches for developing the metamodel is using ML methods. Based on training data or past experience, inferences can be made using the theory of statistics within the ML model [38]. Therefore, one can expect that by learning the correlations between input variables and outputs obtained from FEA models, the ML model can provide an algorithmic solution for the inferences. This can be perceived as a supervised learning problem [39]. Furthermore, given that the structural performance of wind turbine foundations by FEA is assessed through multiple indicators, e.g., ultimate failure moment, yield moment, cracking moment, etc., a multi-output regression problem will be applied because the developed ML models are expected to map input variables into an array of continuous outputs.

The objective of the multi-output regression is to simultaneously predict multiple continuous output variables. Borchani et al. [40] described two general methods that can be applied to multi-output regression, namely problem transformation methods, and algorithm adaption methods. Problem transformation methods can be used to convert multi-output regression problems into multiple single-output regression problems. However, the main shortcoming of conducting these methods is that dependencies between multiple outputs may be neglected, thus influencing the eventual predictive performance. Therefore, Pyromitros-Xioufis et al. [41] introduced several extended problem transformation methods to solve the aforementioned problem, including multi-target regressor stacking and regressor chains. Nonetheless, applying these problem transformation methods will result in less desirable predictive performance and more computational complexity [40].

One widely used algorithm adaption method is Multi-target Regression Trees (MTRT), which enables the simultaneous prediction of numerical multi-outputs using regression trees [42]. Besides, in order to enhance the predictive performance, Random Forest (RF) can be applied by ensembling a set of MTRT models to combine and average predictions of models [43–45]. Li et al. [46] developed a multi-output RF model to predict structural damages and demonstrated that RF is a promising ML method for this multi-output problem.

Artificial Neural Network (ANN) has also shown to have potentials to deal with the multi-output regression problems [47–49]. For instance, Feedforward Neural Network (FFNN) has been adopted broadly in previous studies. Zhao et al. [50] proposed an approach using the multi-output FFNN model to interpret the real-time data during the tunnel construction process. In this research, the developed FFNN model can accurately predict 7 types of mechanical indexes, representing the real-time geological conditions of the tunnel, thus enabling safety assurance during the excavation.

So, it can be posited that both RF and FFNN can potentially deal with the complex multi-output problems that suites the domain of structural analysis of built assets. Therefore, this research will explore both algorithms and compare their performances.

2.4. Metaheuristic optimization of metamodels

The development of the ML model requires frequent fine-tuning of hyperparameters of the model until the best predictive performance is achieved [51]. Studies have also shown that the application of metaheuristic methods can contribute to the optimization of hyperparameters with better efficiency and effectiveness in searching (near-) optimum solution in a great range of problems and avoiding problems that may occur in the traditional optimization algorithms [52]. Metaheuristic optimization can also be implemented on the input variables involved in the ML model to eliminate irrelevant or redundant input variables. Metaheuristic methods mainly include evolutionary algorithms such as Genetic Algorithms (GA), Artificial Immune Systems (AIS), Genetic Programming (GP), or other methods such as Particle Swarm Optimization (PSO).

Among different metaheuristic methods, GA is effective for integration into the ML model to optimize the hyperparameters [53,54]. GA is an approach based on the theory of natural selection to find approximately optimal solutions for optimization problems. In GA, a population consisting of a set of individuals will be generated, called chromosomes, where different gene fragments will represent a solution to the problem. Individuals in the population will be evaluated based on the fitness function and a natural selection will be applied to select individuals having better phenotypic characteristics for reproduction, while more competitive individuals will be eliminated. Besides, several genetic operators are involved in GA to extend the searching space and maintain the diversity of the population to be evaluated, including the crossover and mutation [52]. Individuals selected after the selection stage will be modified in the stage of reproduction employing crossover and mutation to generate new populations, called the offspring. Therefore, individuals in the offspring will be evaluated and selected repetitively to generate new offspring to start a new iteration, until the stopping criteria are satisfied [52].

Oliveira et al. [54] proposed a GA-based method for simultaneously optimizing the selection of input features and optimizing the setting of learning parameters and hyperparameters of three typical types of ML models, namely SVM, ANN, and decision trees. Therefore, in this study, a GA approach will be adopted to optimize the performance of ML models and also to find the optimum design alternatives.

3. A metamodel-based generative design framework

In this study, a generative design framework is proposed using the metamodeling technique and GA. The main objective of this framework is to explore the automatically generated design alternatives rapidly and accurately and obtain the design optimum that balances the cost and structural performance. Fig. 2 shows an overview of the proposed framework, which mainly consists of three stages, namely database establishment, the metamodel development, and the generative design development.

Overall, the database establishment stage is dedicated to building the necessary dataset for the development of the metamodel based on the FEA model. It should be highlighted that this step is only needed if either the available historical data is insufficient or no data is available. In case enough data is already available, this step can be skipped. The second stage, i.e., the metamodel development, aims at building a surrogate to replace the FEA simulation performed on the wind turbine foundation design for obtaining the corresponding structural performance, using the ML techniques to correlate the design parametric and corresponding structural performance. Therefore, the developed metamodel can be expected to enable the fast evaluation of each generated design solution in this generative design process. The third stage of the framework explores and evaluates the design options under the given constraints and objectives, and evolve the design solutions using GA until the convergence criteria are satisfied to provide optimal design solutions.

3.1. The dataset establishment

This stage aims to establish the dataset required for the ML model development. As mentioned in Section 3, this stage is only needed to complement or generate the necessary data for ML in case the data is insufficient or nonexistent. In case enough historical data are available, then this stage can be skipped.

The required dataset contains an array of design solutions, which will provide various design variables as input features in the developing phase of ML models and related structural performance of each design as outputs. To determine which input variables are most influential in the design of wind turbine foundations, 5 interviews with expert structural engineers were held. This approach has been previously applied in other research where ML was used to solve engineering problems [55]. Besides, a similar study conducted by Nicholson [8] has been used as the reference, and eventually, 9 design variables were selected as shown in Table 1. For simplicity, only the circular wind turbine foundation with a slope is considered in this research, as shown in Fig. 3. This is because this type of foundation is the most cost-efficient choice which can result in considerable material saving without compromising the structural performance [11]. However, the proposed method can be easily extended to other types of foundations as well.

The design variables that can be used to define the geometry of the circular wind turbine foundations include the diameter of the foundation base, the thickness of the foundation that can be divided into the height of the slope and the height of the foundation at its outer edge. In addition, the reinforcement configuration will also be included as the

Table 1
Selected design variables in the wind turbine foundation design.

Feature	Description
Base Diameter (BD)	The diameter of the base of the foundation
Anchor Count (AC)	The number of anchors within the foundation
Vertical Load Factor (VLF)	The load factor will be applied to calculating the vertical load transferred from the tower
Base Slope Height (BLH)	The height of the slope part of the foundation
Base Side Height (BDH)	The height of the foundation outer edge which is perpendicular to the base
Top Rad Ratio (TRR)	The reinforcement ratio of the radial reinforcement at the top part of the foundation
Bottom Rad Ratio (BRR)	The reinforcement ratio of the radial reinforcement at the bottom part of the foundation
Top Tan Ratio (TTR)	The reinforcement ratio of the tangential reinforcement in the top part of the foundation
Bottom Tan Ratio (BTR)	The reinforcement ratio of the tangential reinforcement in the top part of the foundation

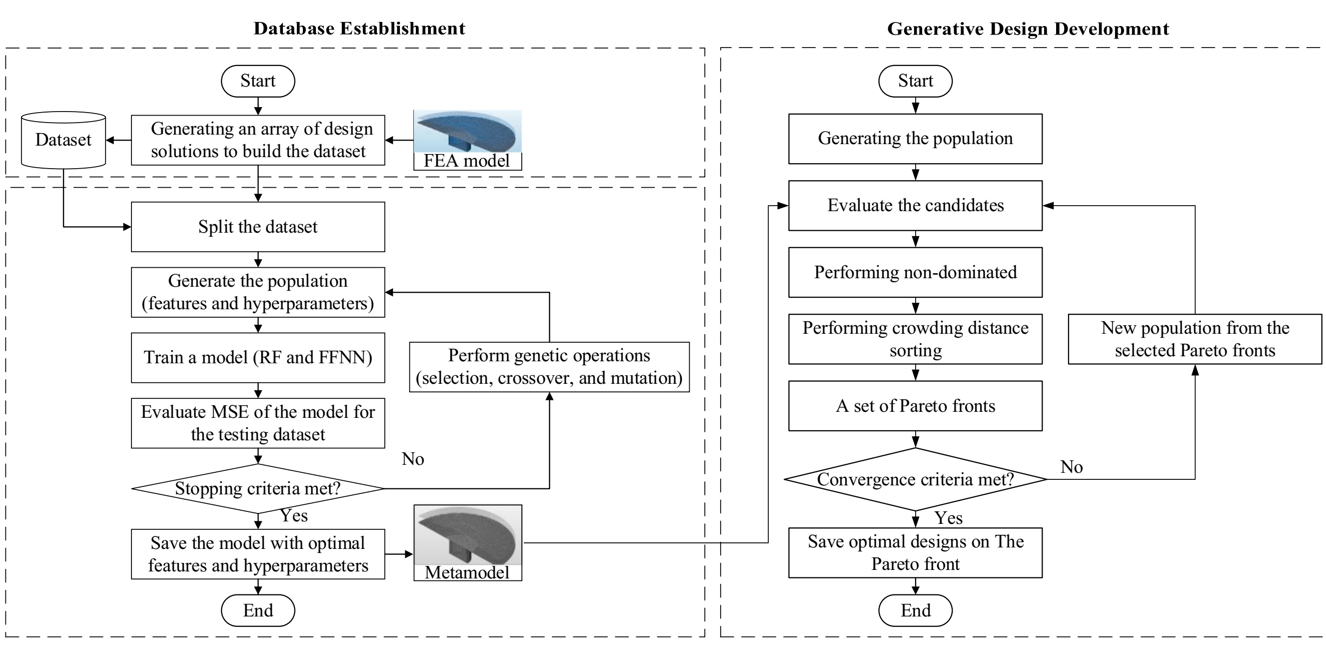

Fig. 2. The framework of the meta-model-based generative design development.

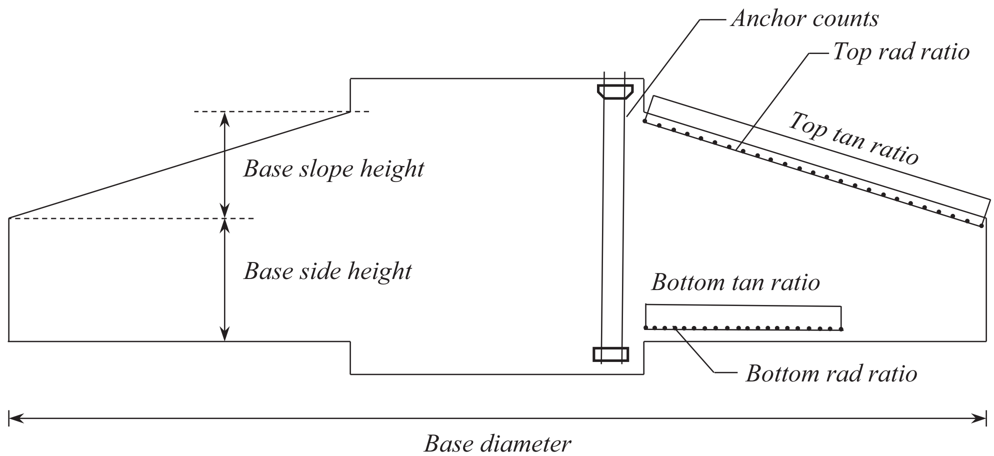

Fig. 3. The schematic diagram of the wind turbine foundation and corresponding input features.

reinforcement ratio of different reinforcement groups. Other design variables include the number of anchors that will be used to anchor the turbine, as well as the applied load factor that will determine the load transmitted from the tower.

Given that the overturning moment is one of the most decisive load cases applied to the wind turbine foundation, the moment-rotation behavior of the wind turbine foundation was selected as the main structural outcome, which can be portrayed by the diagram shown in Fig. 4 [7,8]. As shown in this figure, four stages can be distinguished. The first stage is the cracking stage, where cracks of the reinforcement will occur at P_c. Then, the reinforcement starts yielding at the point P_y, when the stress of the reinforcement reaches its yield strength. As the applied rotation continues to increase, the reinforcement will reach its ultimate strength, where a certain reinforcement group will fail at P_i. Lastly, a yield mechanism will be formed at P_u, where the resistance against further rotation will stop increasing and the foundation reaches its ultimate bending moment capacity.

Considering the difficulty in predicting the complete moment-rotation diagram using the ML method, it is essential to simplify the diagram only using the four key points indicated in Table 2.

To this end, first, a set of n solutions are generated in terms of the design paramours shown in Table 1. Each set of parameters represents a design solution, as shown in Fig. 5. Each time FEA analysis is conducted on a potential design (i.e., a set of design variables), the four key points from the moment-rotation behavior are obtained. These design variables and the moment-rotation key points are integrated as a single data point. After repeating this process for n times, the database needed for the ML is generated in terms of the matrix shown in Fig. 5.

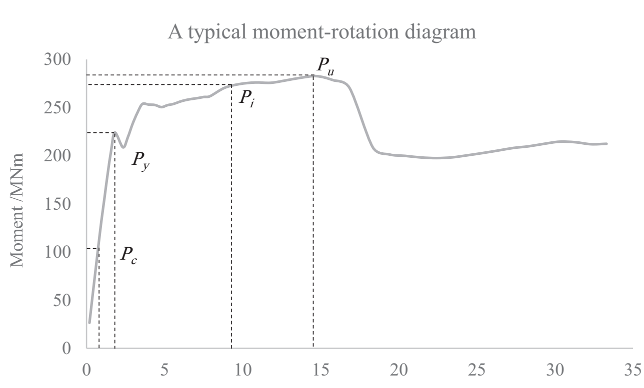

Fig. 4. A typical moment-rotation diagram.

Table 2
Explanation of key points in the moment-rotation diagram.

Point in the diagram	Corresponding outputs	Description
Initial cracking point (P_c)	Initial cracking moment and rotation	The point where the reinforcement starts cracking
Initial yield point (P_y)	Initial yield moment and rotation	The point where the reinforcement starts yielding
Initial failure point (P_i)	Initial failure moment and rotation	The point where the reinforcement starts reaching the ultimate strain
Failure point (P_u)	Failure moment and rotation	The point the resistance against further rotation stops increasing

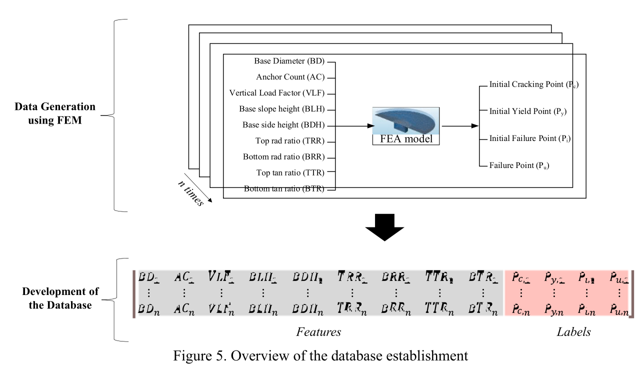

Fig. 5. Overview of the database establishment.

3.2. Machine learning models development

In this stage of the framework, ML methods will be used to develop predictive models that can directly handle multi-output data. ML algorithms selected in this stage include RF and the multi-output FFNN. In addition, several univariate regressions corresponding to every single output will also be studied, as the performance benchmark. This can be done by developing an ML model for each output in the dataset.

Besides, a GA-based optimization approach is adopted for the feature selection during the model training phase and the optimization of hyperparameters, and the k-fold cross-validation method is applied to avoid the overfitting issue.

The bottom left box in Fig. 2 provides an overview of the ML developing process, which consists of the following steps: (1) *splitting the dataset*: the dataset established in the previous phase will be split into the training subset and the testing subset using the k-fold cross-validation method; (2) *generating the gene population*: a population of genes will be generated consisting of the information of feature selection and hyperparameter values and converted from genotypes into phenotypes; (3) *training the models:* both the training subset and the testing subset will be modified based on the feature selection results to train and ML learning models; (4) *evaluating the fitness*: the fitness of each individuals will be calculated; (5) *assessing termination criteria:* if the stopping criteria are satisfied, the process will stop at the current iteration and the optimal features and hyperparameters will be registered, otherwise the next generation will be proceeded using genetic operations; and (6) *performing genetic operations*: the new population will be generated after the crossover and mutation.

k-fold cross-validation is used to ensure the generalizability of the ML model and to prevent the overfitting problem, without making the model sensitive to the data distribution. Using this strategy, the original dataset will be randomly divided into k disjoint folds of approximately same size. Among them, k-1 folds will be used as the training set to enable the self-learning process of the ML model, while the remaining fold will be used as the test set [56]. Then, by averaging performance metrics obtained in the process, the final performance of the model can be calculated.

The selection of the value of k depends on the size of the dataset, where the higher the value of k is, the longer the computational time will require. In this study, 10-fold cross-validation will be performed.

The optimization of ML models will be realized by applying the GA-based approach. After obtaining the splitting of the original dataset, a population will be generated based on input features involved in this dataset and the type of ML algorithm. Subsequently, genotypes of each chromosome will be converted to phenotypes, such as the information about whether a certain feature is selected or reduced and value for one hyperparameter. Then, based on phenotypes from the feature gene fragments, the original dataset will reduce corresponding features. Using hyperparameters defined by the chromosome, the training and testing procedures will start, and the fitness of each chromosome representing parameters and selected features will be evaluated. The whole process will be terminated when the termination criteria are satisfied. Otherwise, individuals with better fitness will be selected using the roulette wheel selection operator, and new solutions are generated using the crossover operator and the mutation operator, until the termination criteria are triggered, or the maximum number of generations reach the pre-defined limit.

The chromosome in each individual within the population should contain phenotypic characteristics about whether a certain feature will be selected or reduced and values of hyperparameters of the ML algorithm. Therefore, the chromosome will be divided into two parts as shown in Fig. 6, of which the first part represents phenotypic characteristics about input features and the second part represents phenotypic characteristics about hyperparameters, using the binary coding system.

Specifically, each element in the first part of the chromosome represents one certain feature. If the value of this element is "0", it means that the corresponding feature will be reduced. On the contrary, if the value is "1", it means that the corresponding feature will be selected.

In this study, two ML algorithms are used, namely RF and FFNN. From previous studies, these two ML algorithms have shown the potential to deal with multi-output regression problems, with good predictive performance, computational speed, scalability, ease of use, and extensibility [40,48]. Specifically, RF can provide good performance in working with large datasets, and it has the ability to overcome the overfitting problem and outliers [57,58]. Besides, another advantage of RF is its interpretability because the feature importance can be generated automatically. In addition, FFNN can also provide good performance by automatically adjusting the model complexity based on the failure history, and most importantly, the ability to learn and represent both the linear and non-linear relationships from the dataset [59].

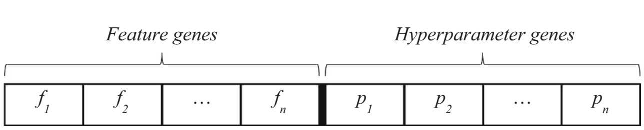

Fig. 6. The structure of the chromosome.

It is worth noting that ML methods are shown to be sensitive to the dimensionality of the problem, i.e., the ratio of features (p) to data points (n) [60,61]. Although ensemble methods (e.g., RF) are commonly believed to outperform other conventional ML methods for high-dimensional problems [62–64], essentially the performance of any ML heavily depends on the nature of the data. Therefore, the performance needs to be assessed on a case-by-case basis [60,61]. Here, the optimization and fine-tuning of hyperparameters are very important to boost the performance of the model, especially for high dimension problems [55,60,61,65].

Table 3 lists all the hyperparameters which will be optimized using this GA method. Each hyperparameter will be linked to a certain fragment within the second part of the chromosome. Subsequently, the binary code for each chromosome fragment will be converted to a decimal value as the real value of each hyperparameter.

The fitness function measures the performance of each individual after the evaluation, which can also be used as a metric for stopping the iterative process early when a specific value of the fitness is achieved [66]. Several performance evaluation metrics have been introduced by Borchani et al. [40] for multi-output regression problems. In this study, the mean squared error (MSE) will be applied in the fitness function. According to Urbanek et al. [67], MSE can be regarded as the best fitness function because of its statistical properties. Besides, using MSE as the metric can ensure that trained models with outlier predictions will be eliminated, given the fact that the MSE will magnify those huge errors.

Therefore, in the proposed GA-based model optimization approach, the fitness function used for the selection stage will be MSE, as indicated in Eq. (1), according to [40].

$$fitness = MSE = \sum_{i=1}^{d} \frac{1}{N_{test}} \sum_{l=1}^{N_{test}} \left(y_i^{(l)} - \hat{y}_i^{(l)} \right)^2 \tag{1}$$

Where:

D: the number of outputs

N_{test}: the size of the test dataset

$y_i^{(l)}$ the actual value of the i^{th} output corresponding to $X^{(l)}$ in the test dataset

$\hat{y}_i^{(l)}$ represents the predicted value of the i^{th} output corresponding to $X^{(l)}$.

However, it is worth noting that the calculation of MSE in the multi-output context may result in distinct ranges because the predictive performance of each output is calculated separately. Therefore, it is critical to normalize each error value, instead of simply calculating the average value. This can be done by normalizing each output before the

Table 3
Explanation of hyperparameters required to be optimized.

ML method	Hyperparameter	Description
Random Forest (RF)	n_estimators	The number of trees in the forest
	max_depth	The maximum depth of each MTRT in the forest
	min_samples_split	The minimum number of samples needed to split an internal node
	min_samples_leaf	The minimum number of samples required to be at a leaf node
Feedforward Neural Network (FFNN)	n_layers	The number of hidden layers
	n_nodes	The number of nodes within each hidden layer
	epochs	The number of training epochs
	learning_rate	The learning rate of the backpropagation algorithm
	batch_size	The number of training samples used in one iteration

model training phase. Therefore, for every output, the range of potentially calculated MSE will between 0 and 1, while the possible MSE for all eight outputs will range from 0 to 8. The smaller the value of MSE is, the more accurate prediction can be achieved.

3.3. Generative design development

The main target of this stage, which is shown on the right box in Fig. 2, is to realize the generative design of the wind turbine foundations by applying the developed metamodel to enable the rapid and accurate evaluation of each generated design alternative and using GA to achieve the evolution of the generated design solutions to further increase the diversity in the solution space that will be explored and assessed. Specifically, in this study, the generative design is essentially intended to solve a design optimization problem, given the specific constraints and objectives.

The process will start with the generation of an array of design solutions, which only include the input design variables that have been selected after the feature selection phase in the metamodel development. Afterward, these input design variables will be assessed and filtered according to the pre-defined constraints. In this study, three constraints will be applied to the diameters of rebars from different reinforcement groups: (1) for the radial reinforcement, the diameter of the rebar located in the top area of the foundation will not be larger than the rebar located in the bottom area; (2) for the tangential reinforcement, the diameter of the rebar located in the top area of the foundation will not be larger than the rebar located in the bottom area; (3) the slope of the foundation is 1:4 unless this would lead to a Base Side Height less than 1.5 m, and in that case, the value of feature Base Side Height will be 1.5 m. In addition to these three constraints, the value of each design variable will also be confined to a certain value range, which will be determined according to different cases.

Subsequently, the generated design solutions will be evaluated against two indicators, namely the structural performance and the cost of each design solution. The evaluation of the former can be done by using the developed metamodel, which can predict the eight structural performance outputs simultaneously. However, in this study, only the ultimate bending moment will be selected as the indicator for determining the structural capacity of the design, because the ultimate bending moment capacity is the critical consideration to ensure the resistance of the structure against the overturn resulted from the significant bending moment owing to the aerodynamic interaction. Besides, the cost of each design solution will also be calculated, which is mainly dominated by the volume of the concrete in the cylindrical and conical parts of the foundation and the weight of the radial and tangential reinforcement. Therefore, a multi-objective optimization problem can be formulated as indicated in Eq. (2), which aims at finding the optimal design solution with the lowest cost and the highest ultimate bending moment capacity:

$$\mathrm{f}_1 = maxB(x) \tag{2}$$

$$\mathrm{f}_2 = minC(x) \tag{3}$$

Subject to:

$$g_1(x) = x_{top\ radial\ rebar} - x_{bottom\ radial\ rebar} < 0$$

$$g_2(x) = x_{top\ tangential\ rebar} - x_{bottom\ tangential\ rebar} < 0$$

$$g_3(x) = x_{Base\_slope\_height} \geq 1.5$$

$$x_i^L < x_i < x_i^U, i = 1, \ldots, N$$

Where:

x: the set of input design variables from a design solution

$B(x)$: the objective function aiming at obtaining the maximum bending moment

$C(x)$: the objective function determining the cost of the design

$g_1(x)$, $g_2(x)$, and $g_3(x)$: constraints,

x_i^L: the lower variable boundaries of the design variable x_i

x_i^U: the upper variable boundaries of the design variable x_i

N:total number of the input design variables.

The solution to this multi-objective optimization is a Pareto front that represents a set of design alternatives that are non-dominated to each other. The Pareto optima can also represent the trade-off between the two objectives and enable the decision-makers to select the most favorable result, thus it can be more practical for real-life problems. In this stage, the Fast Non-dominated Sorting Genetic Algorithm (NSGA-II) will be applied to generate the Pareto front concerning the cost and the structural performances, considering its efficiency and wide application in civil engineering for solving multi-optimization problems [68–70].

NSGA-II defines the fitness and the probability of being selected based on the Pareto dominance. In order to rank each solution on the Pareto front regarding the performance, the crowding distance will be applied to measure the closeness of each design solution to the other two design solutions in its neighborhood [71]. By using the tournament selection method, two design solutions will be randomly compared and the design solutions with higher crowding distance will be selected [72].

Compared to the chromosome design containing both the feature genes and hyperparameter genes, the chromosome used in this stage will only include the feature genes after the feature selection. Therefore, an initial population will be generated, in which each individual will be represented by the binary-coded chromosome, and the offspring will be created after performing the crossover and mutation. After combining the initial population and its offspring, the non-dominated sorting will be applied to the combination and several non-dominated solutions will be identified. And by applying the tournament selection based on the calculated crowding distance of each individual on the non-dominated fronts, the parents can be selected from the combination and after performing the crossover and mutation, the new offspring can be created to ensure the diversity of the population.

By repeating the abovementioned process in every iteration until the convergence criteria are satisfied, the Pareto optima can be obtained. In addition, the concept of elitism will also be utilized in the approach, which can accelerate the convergence speed of GA and also ensure that the individuals with good performance will not be reduced in the iterative process [71].

4. Case study

This section demonstrates the results of a case study conducted to validate the proposed framework. The case study contains two phases. In the first phase, a metamodel was developed based on the dataset consisting of 4971 design solutions of wind turbine foundations. In the second phase, a generative design approach was adopted using this metamodel to realize the rapid evaluation of the structural performance of each design solution.

The dataset used in this research is provided by WindBase [73], which is a Dutch design company with over 25 years of experience in wind turbine foundation designs. To ensure the consistency of the data structure, following Table 1, this dataset only contained the range of

Table 4
Input variables from the dataset.

Variable	Range	Step
Base Diameter (BD)	12.5 m ~ 25 m	2.5 m
Anchor Count (AC)	60– 120	30
Vertical Load Factor (VLF)	0.9– 1.35	0.45
Ratio	0.12– 0.18	0.02
Top Rad Ratio (TRR)	20 mm, 25 mm, 32 mm	N/A
Bottom Rad Ratio (BRR)	20 mm, 25 mm, 32 mm	N/A
Top Tan Ratio (TTR)	20 mm, 25 mm, 32 mm	N/A
Bottom Tan Ratio (BTR)	20 mm, 25 mm, 32 mm	N/A

variables shown in Table 4. Based on the input variables represented in Table 4, the values of several input features, such as Base Diameter, Anchor Count, and Vertical Load Factor, can be directly obtained, while the rest of input features can be calculated based on corresponding variables shown in the table. Specifically, the variable Ratio in Table 4 stands for the ratio of the total thickness of the foundation (i.e., the sum of Base Slope Height and Base Side Height) and the diameter of the foundation. Therefore, the total thickness can be calculated based on Ratio and the fixed slope rate of 1:4 (as explained in Section 3.3), from which Base Slope Height and Base Side Height can be derived. Similarly, based on the diameter of a certain group of reinforcement, the corresponding reinforcement ratio can be calculated as the input features.

Table 5 shows an example of how the input and outputs of WindBase look like. As shown in this Table 5, the input values shown in Fig. 5 and Table 1 provided to WindBase and the moment-rotation key points were exported from the results of the FEA analysis previously performed by WindBase.

Parameters for the GA process in the metamodel development stage were pre-defined, as indicated in Table 6. In this case study, the size of the initial population was set to 100, which means 100 individuals were used as the input of the GA process. Besides, during the crossover stage, the crossover rate was set to 0.8, which represents the possibility of two randomly selected individuals after the selection of exchanging gene fragments. Furthermore, the possibility of the mutation occurring in each offspring generated after the crossover phase was 0.2. The stopping criterion was the maximum generation number of 50.

4.1. Metamodel development

As explained in Section 3.2, MSE was selected as the fitness function to evaluate the competitiveness of each generated individual. Given the fact that the usage of the MSE can provide a physical meaning and is grounded in reasonable probabilistic assumptions, it was used as the main metric to indicate the model's predictive performance. However, MSE cannot intuitively indicate how accurate the developed ML model was. Therefore, another performance metric named PRED(25) was used in this research, which represents the percentage of predictions falling within 25% of the true value [54,74]. Therefore, the overall predictive performance of multi-output models was determined by MSE and PRED (25). The former was calculated by summing up the MSE obtained on each output as indicated in Eq. (1), while the latter was the average of the PRED(25) achieved on each output. Unlike MSE, a higher value achieved in PRED(25) represents a higher accuracy of the prediction.

Table 7 indicates the result after testing GA-based ML models using this dataset. Furthermore, to provide a more intuitive representation of results, regression plots for each multi-output model are shown in Figs. 7 and 8. It is worth noting that for the multi-output RF and FFNN, not only the overall predictive performance is given, but also the performance on each output. Besides, for every single output, two single-output ML models were also developed using RF and FFNN respectively. The only difference between the multi-output regression models and the single-output models is that the former can predict 8 outputs, while the latter predict one specific output independently.

Table 8 represents the configuration of obtained optimal multi-output models using these two ML algorithms. In terms of the feature selection, the two models applied different feature combinations. For the multi-output RF model, the feature Anchor Count was reduced, while there is no feature that has been reduced in the multi-output FFNN model.

It is clear that two multi-output ML models can provide the same predictive performance regarding the calculated MSE. However, the calculated PRED(25) of the multi-output RF is considerably higher than the multi-output FFNN. Besides, based on these results, the multi-output RF model has a more desirable performance in predicting all the outputs,

Table 6
Pre-defined GA parameters.

GA parameter	Description	Value
Population size	The number of individuals that will be evaluated and selected	100
Offspring size	The number of individuals which will be re-generated after one iteration	100
Crossover rate	The probability of two random individuals after selection replacing their gene fragments	0.8
Mutation rate	The possibility of the occurrence of mutation	0.2
Number of generation	The number of iteration in the GA process	50

Table 7
Results obtained from the case study.

Output	Metric	Multi-output RF	Single-output RFs	Multi-output FFNN	Single-output FFNNS
Overall	MSE (normalized)	0.0169	0.0161	0.0168	0.0201
	PRED(25)	93.26%	94.82%	90.80%	91.07%
Initial cracking rotation (mrad)	MSE (normalized)	0.0019	0.0017	0.0019	0.0019
	PRED(25)	91.97%	92.78%	81.11%	84.63%
Initial cracking moment (MNm)	MSE (normalized)	0.0013	0.0013	0.0014	0.0014
	PRED(25)	93.64%	94.89%	91.29%	91.65%
Initial yield rotation (mrad)	MSE (normalized)	0.0025	0.0025	0.0026	0.0027
	PRED(25)	91.41%	92.60%	83.87%	85.09%
Initial yield moment (MNm)	MSE (normalized)	0.0014	0.0014	0.0015	0.0015
	PRED(25)	95.31%	96.60%	95.49%	95.88%
Initial failure rotation (mrad)	MSE (normalized)	0.0009	0.0006	0.0008	0.0010
	PRED(25)	92.82%	95.68%	89.56%	86.95%
Initial failure moment (MNm)	MSE (normalized)	0.0014	0.0012	0.0016	0.0015
	PRED(25)	95.63%	97.12%	96.94%	97.34%
Failure rotation (mrad)	MSE (normalized)	0.0061	0.0063	0.0059	0.0089
	PRED(25)	91.95%	91.67%	90.94%	89.29%
Failure moment (MNm)	MSE (normalized)	0.0013	0.0011	0.0012	0.0012
	PRED(25)	96.22%	97.24%	97.28%	97.83%

Table 5
An example of the dataset provided by WindBase.

Solutions	Input design variables									Outcomes			
	BD (m)	AC (unit)	VLF (ratio)	BLH (m)	BDH (m)	TRR (ratio)	BRR (ratio)	TTR (ratio)	BTR (ratio)	Pc (mrad, MNm)	Py (mrad, MNm)	Pi (mrad, MNm)	Pu (mrad, MNm)
Solution 1	17.5	90	0.9	0.6	1.5	0.206%	0.206%	0. 0997%	0. 0997%	(0.81, 69.89)	(1.03, 85.28)	(5.24, 137.00)	(14.88, 140.39)
Solution 2	12.5	90	0.9	0	1.5	0.288%	0.288%	0.1396%	0.1396%	(1.50, 44.26)	(2.11, 60.71)	(6.11, 95.85)	(12.23, 96.29)
⁝	⁝	⁝	⁝	⁝	⁝	⁝	⁝	⁝	⁝	⁝	⁝	⁝	⁝

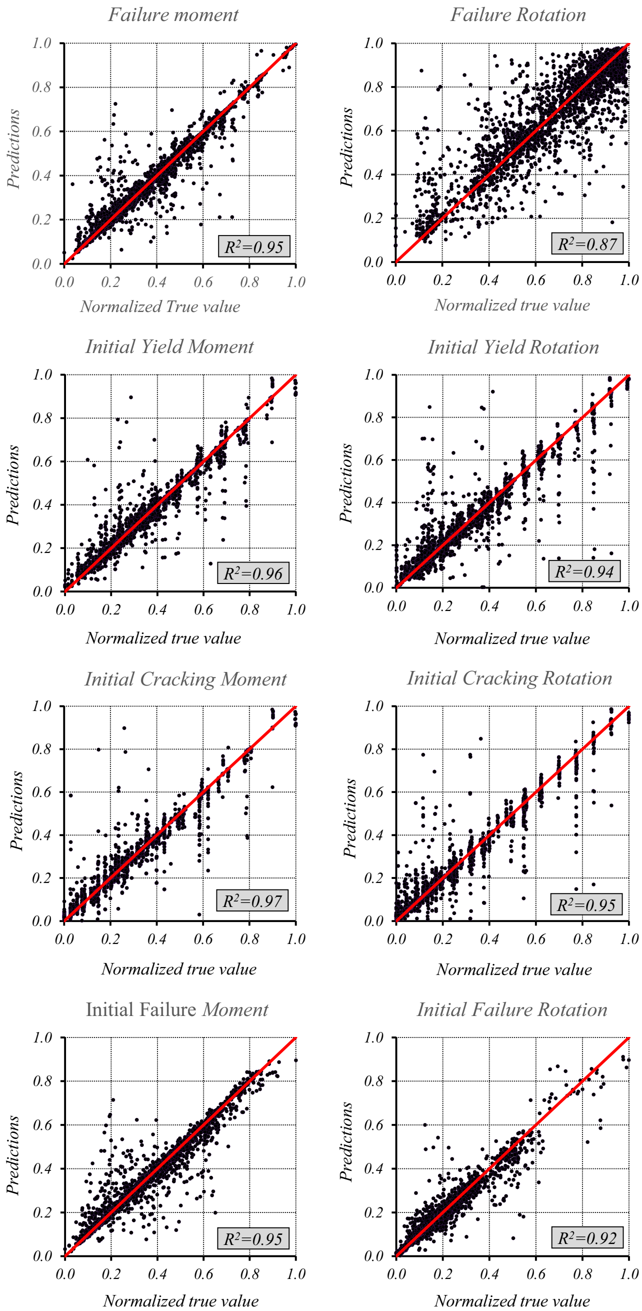

Fig. 7. Regression plots of the multi-output RF model.

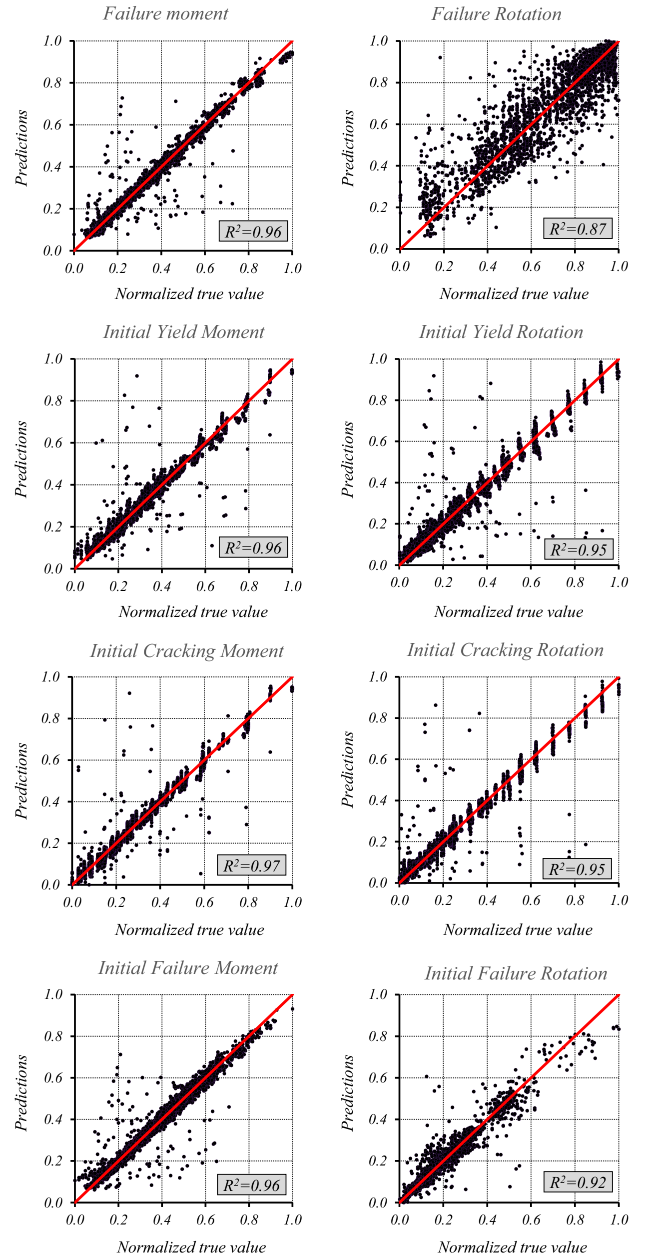

Fig. 8. Regression plots of the multi-output FFNN model.

while the multi-output FFNN has a poorer predictive performance for the initial cracking rotation and initial yield rotation. Therefore, the multi-output RF model can be regarded as the more desirable choice that can provide better predictive performance in terms of predicting all eight outputs.

Furthermore, Figs. 9 and 10 indicate the regression plots of each output predicted by corresponding single-output models. When comparing the two modeling approaches, it appears that single-output models have a slight edge over the multi-output model (i.e., an average of 0.82% higher prediction accuracy). Therefore, it is recommended to use the single-output ML models when only one specific design output is sought. However, when it comes to the prediction of all 8 outputs simultaneously, considering that the running time of an array of single-output ML models is longer than only using one multi-output model, as shown in Table 9, and also given the relatively negligible difference in the prediction accuracy, the multi-output model can be regarded as a more desirable choice, especially when considering the extension of the application of developed metamodels to generative design.

4.1.1. Analysis of feature importance

After developing ML models in this case study, it was also crucial to obtain a better understanding of these models regarding their interpretability, which explains the behavior of models in the entire population, especially when these ML models are normally regarded as "black box" tools.

One of the most effective methods to investigate the interpretability of models is conducting a sensitivity analysis to examine how the influence of each input feature can affect the predictive performance of models. In this study, permutations were used to randomly shuffle a single feature value. Feature importance can be obtained based on the

Table 8
Results of the feature selection and optimal model configurations.

Machine learning algorithm	Reduced feature	Hyperparameter	Value
RF	Anchor Count (AC)	n_estimators	380
		max_depth	48
		min_samples_leaf	2
		min_samples_split	3
FFNN	None	n_layers	4
		n_nodes[a]	17–29–31–16
		epochs	500
		learning_rate	0.015
		batch_size	10

[a] The structure of the value of hyperparameter n_nodes means there are 17 nodes, 29 nodes, 31 nodes, and 16 nodes in the first, the second, the third, and the fourth hidden layer respectively.

change in the model score [75]. It is worth noting that the feature importance obtained using permutation cannot reflect the inherent predictive value of features, but only the importance of features for a particular model. In addition, when using RF, it is also possible to adopt another feature importance measurement method, called the Mean Decrease Impurity (MDI) or Gini importance, which obtains each feature importance by summing the number of splits included in the feature, and weighted by the number of samples split [76]. Therefore, the relative impurity-based feature importance was also calculated for the comparison.

Because in the previous section results indicated that multi-output RF is a more favorable choice, only the multi-output RF model was used to conduct this sensitivity analysis. It is worth noting that the selected model has reduced the feature Anchor Count during the development.

Fig. 11(a) shows the results of the permutation-based sensitivity analysis on the optimal multi-output RF model, where feature Anchor Count has been reduced from the model training phase. In addition, the impurity-based feature importance is indicated in Fig. 11(b). The results of the impurity-based feature importance match the results obtained from the sensitivity analysis, except the rank of the feature Vertical Load Factor. Therefore, for this optimized multi-output RF model, features including Base Slope Height, Base Diameter, Bottom Tan Ratio, and Base Side Height were the most important features that were more decisive in influencing the moment-rotation behavior of wind turbine foundations, while Top Rad Ratio had the least importance.

4.2. Generative design

In this section, a framework for realizing the generative design of the wind turbine foundations was developed and implemented, using the multi-output RF as the evaluating model, whose model configuration can be found in Table 8. The objective of the generative design method was to automatically generate and assess the design solutions under the given constraints with maximum ultimate bending moment capacity and minimum cost. Besides, as explained in Section 3.3, the NSGA-II was performed to obtain the Pareto front that represents the optimal trade-off between the cost and the ultimate bending moment capacity. Furthermore, to provide a more thorough examination of the performance of this generative design model, a comparison was made between the optimal Pareto front obtained from the generative design process and the optimized design solutions derived from the dataset provided by WindBase under the same design constraints.

4.2.1. Configuration of the generative design

Three design constraints were applied as indicated in Eqs. (2) and (3). Besides, in order to use the generative design model to explore a considerable design solution space, the value range of each design

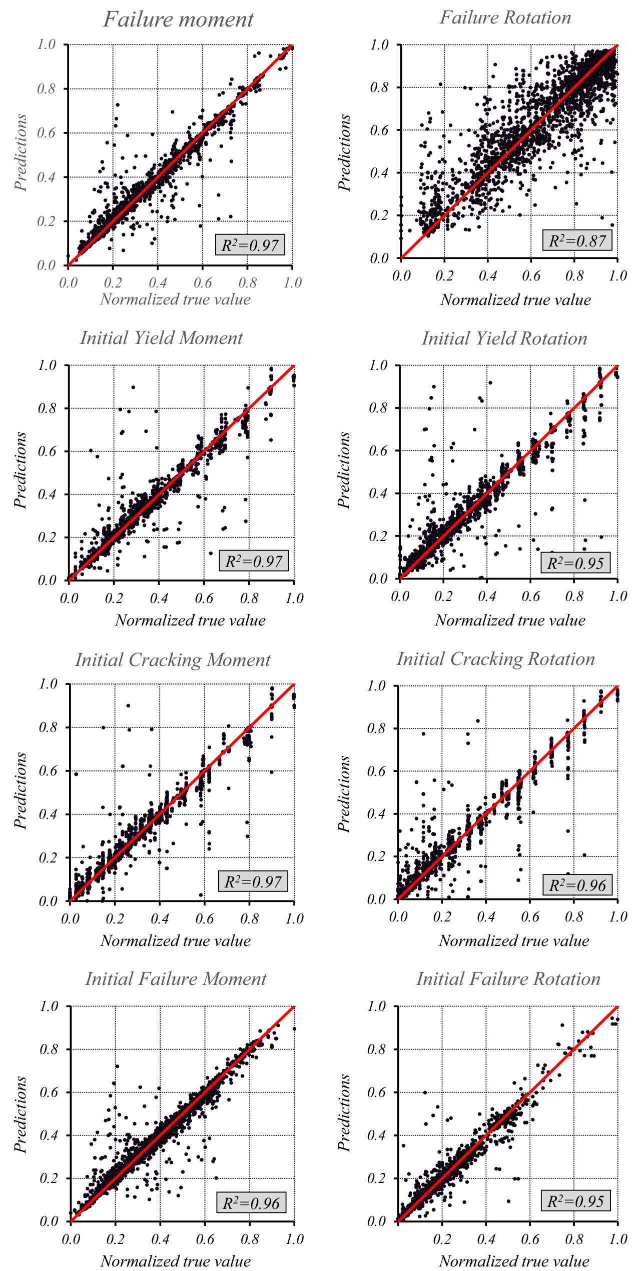

Fig. 9. Regression plots of the single-output RF model.

variable was pre-defined, as shown in Table 10. Because the design variable Anchor Count had been reduced from the multi-output RF model, this design variable was directly replaced by a fixed value. The hyperparameters of the NSGA-II process are shown in Table 11. The stopping criterion applied in this case study was the maximum generation number, which was set to 100.

4.3. Results of the generative design process

To provide a comparison to the results obtained using the proposed generative design method, first, a series of optimal design solutions were derived from the dataset provided by WindBase. These are design solutions developed by designers. All the design solutions in WindBase which falls under the ranges shown in Table 10 were selected. The reinforcement ratios in this sub-dataset were converted into corresponding diameters of rebars, to facilitate the assessment of whether

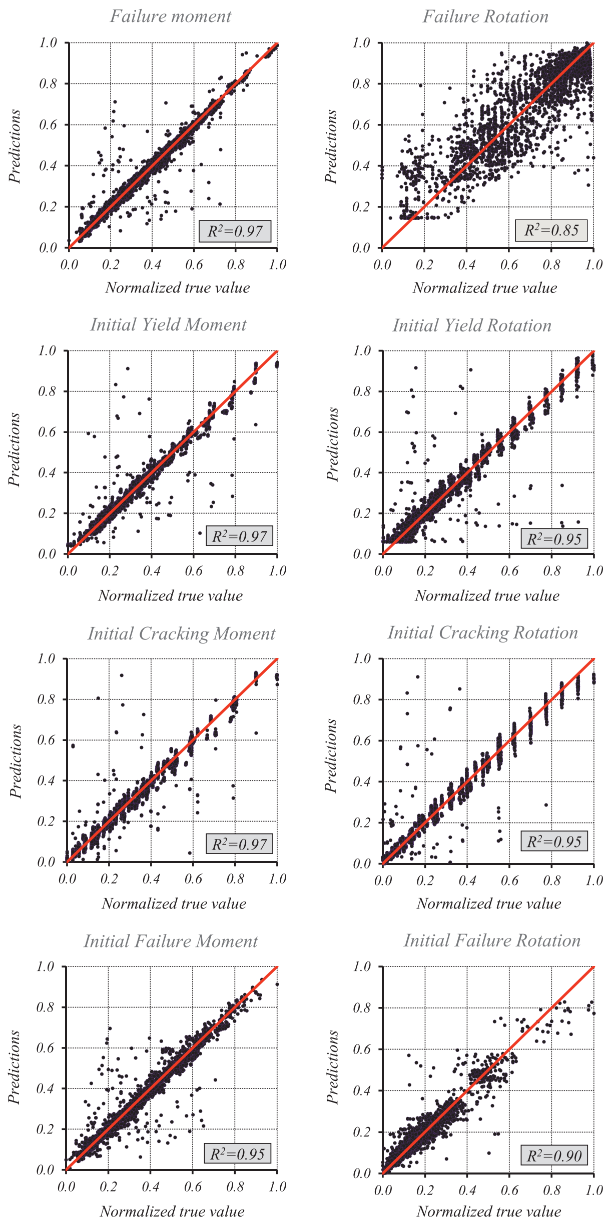

Fig. 10. Regression plots of the single-output FFNN model.

Table 9
The average running time of each model for predicting 8 outputs.

Model	Multi-output RF	Single-output RFs	Multi-output FFNN	Single-output FFNNS
Average computational time	0.12 s	0.73 s	0.08 s	0.59 s

these design solutions satisfy the constraints. The derived optimal design solutions from this sub-dataset can be found in Fig. 12.

Based on the constraints and the pre-defined value ranges of different design variables, the proposed generative design method was implemented. Fig. 13 provides the final results. However, it is difficult to compare the two Pareto fronts directly, because none of them can dominate the other. According to Salimi et al. [77], it is possible to numerically convert and assess the near-optimum solutions using several indicators, thus enabling the comparison. Therefore, a widely applied hypervolume indicator is used in this study [78]. This indicator reflects the exclusive dominated area of a set of near-optimum solutions in the Pareto front to a pre-determined reference point. Pareto fronts with more diversity will have higher values in this indicator.

The reference point selected for conducting the calculation of the hypervolume indicators of each point in the two Pareto fronts is the worst scenarios of the two objectives (i.e., the largest cost and the smallest ultimate bending moment capacity from the two Pareto fronts), as suggested by Bradstreet et al. [79]. Therefore, according to Salimi et al. [77], a reference area can be created by the reference point and its diagonal point (i.e., the point with the smallest cost and the largest ultimate bending moment capacity). The hypervolume indicator can be computed as the ratio of the area bounded by one Pareto front point divided by this reference area, as represented in Table 12. The comparison shows the Pareto front obtained from the generative design can achieve a desirable accuracy with 1.75% percent error, indicating that the proposed method can achieve a good accuracy in deriving the optimized design solutions.

Currently, the design optimization of the wind turbine foundation conducted in Windbase heavily relies on the experience of the engineers, where they will empirically define the initial design variables and fine-tune the values of these variables based on the corresponding output of the structural performance. However, the success of applying this method would highly depend on the efficiency of the adopted evaluation method for calculating the structural performance, as described in the previous section. As shown in Fig. 12, the identification of design optimum is similar to utilizing the grid search, where all the possible solutions were identified and evaluated using FEA, and the optimal Pareto front can be derived accordingly. However, this grid search is not feasible nor realistic in the actual design process of wind turbine foundations, due to the inefficiency of the traditional evaluation method such as FEA. Table 12 also provides a comparison between the utilization of generative design and the traditional FEA-based design optimization on the average computational time for obtaining the optimal design solution of the wind turbine foundation. From the comparison, it demonstrates that the metamodel-based generative design method can significantly reduce the computational time required for the evaluation of the structural performance of each design (i.e., 99.93%), thus enabling the rapid exploration of the larger solution space under the given constraints. Therefore, the results prove that the proposed metamodel-based generative design method is superior to the FEM-based design optimization method in terms of computational speed and the outcome of the obtained design optima.

5. Discussions

The main contribution of the presented study is providing an opportunity to determine how and to what extent the metamodeling technique and the concept of generative design can be used to streamline the wind turbine foundation design process. Based on the results represented in Section 4, this study found that the developed multi-output RF model is the most favorable choice, considering the computational speed and accuracy. Although this finding differs from that of Kayri [78], who found that the neural network can be regarded as the best option for solving big and complex data mining problems, it is broadly consistent with the earlier study conducted by Adusumilli et al. [80], in which RF was examined to have better performance than the neural network under the similar multi-output regression context. One possible reason for the difference in the predictive performance of the two selected multi-output regression models is the strong non-linearity between the inputs and outputs in this problem context. The general method to improve the performance of the FFNN to tackle the computational complexity is to extract uniquely abstract features by increasing the number of hidden layers [81]. However, currently, the maximum of the hidden layers in the FFNN architecture is only 4, resulting in a rather "shallow" neural network to model the complex non-linear relationship,

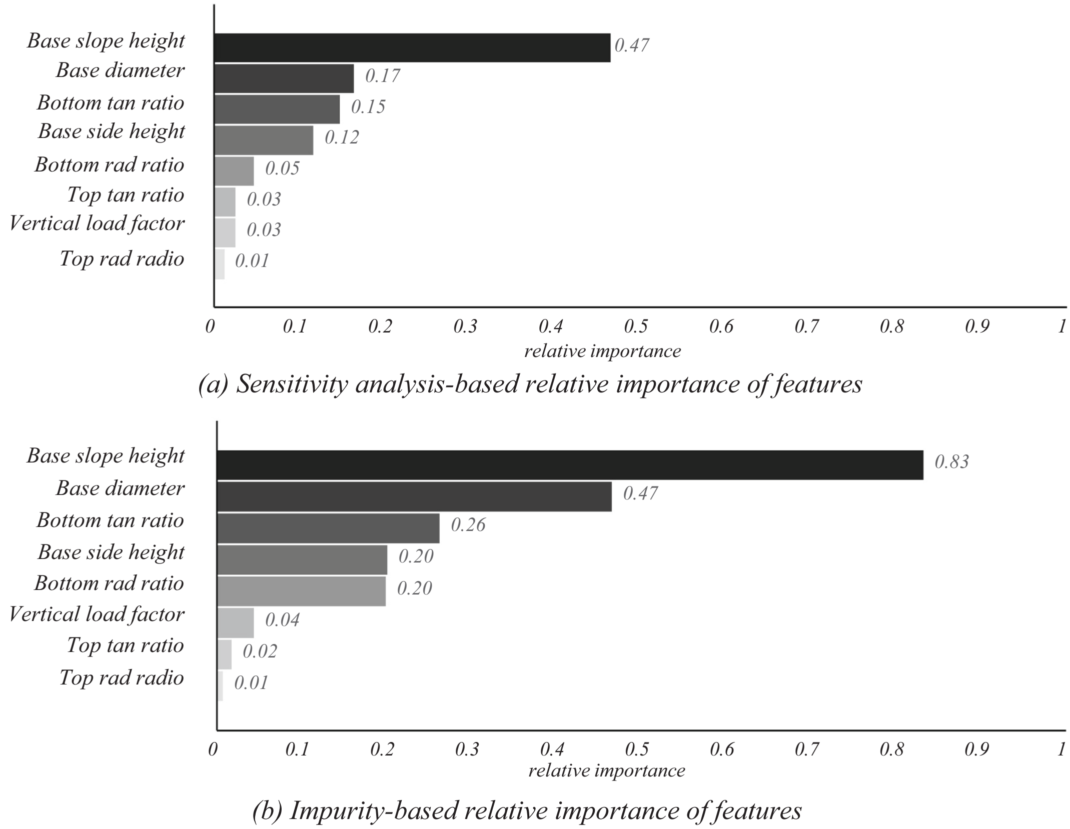

Fig. 11. Feature importance analysis on the multi-output RF model.

Table 10
The list of design variables in the generative design.

Design variable	Range	Step
Base Diameter (BD)	12 m ~ 27.5 mm	0.5 m
Anchor Count (AC)	90 (fixed)	N/A
Vertical Load Factor (VLF)	0.9 (fixed)	N/A
Ratio	0.12– 0.18	0.02
Top Rad Ratio (TRR)	20 mm, 25 mm, 32 mm	N/A
Bottom Rad Ratio (BRR)	20 mm, 25 mm, 32 mm	N/A
Top Tan Ratio (TTR)	20 mm, 25 mm, 32 mm	N/A
Bottom Tan Ratio (BTR)	20 mm, 25 mm, 32 mm	N/A

Table 11
Pre-defined NSGA-II parameters.

NSGA-II parameter	Description	Value
Starting population size	The number of individuals that will be included in the starting population	500
Offspring size	The number of individuals after performing GA operations on the starting population	500
Crossover rate	The probability of selecting two random individuals for crossover operations	0.9
Mutation rate	The possibility of the occurrence of mutation	0.1
Number of generation	The number of iteration in the NSGA-II process	100

while RF shows the better capability to cope with this non-linearity with the same dataset. The reason why the number of hidden layers has been confined to be under 4 in this study is mainly because of the size of the current dataset for training. This is because as the number of hidden layers of the neural network increases, more adjustable model parameters will be involved, thus increasing the degree of freedom, as well as the complexity of the regression model. Consequently, the over-fitting problem will ensue. One widely applied solution is to increase the size of the dataset. Therefore, different results may be expected when the dataset can be extended in the future.

In addition, the model interpretation shows that the diameter of the foundation base, the reinforcement ratio of the tangential reinforcement at the bottom, and the thickness of the foundation side are the most decisive design variables in predicting all the outputs, while the number of anchor, the contributes the least. The finding regarding the relative importance of these design variables is in line with the experiential expectation of experts in WindBase that geometry and tangential reinforcement configuration of the foundation has a significant contribution to the ultimate bending moment capacity. However, to the best of the authors' knowledge, there has been no detailed parametric study investigating the relative importance and influence of these design variables on the ultimate bending moment.

In general, the single-output model offered a slightly better prediction of individual outputs. Therefore, it can be argued that if a single design indicator is sought, it is advantageous to use a single-output model that independently predicts the moment-rotation behavior of the wind turbine foundation. This finding is contrary to that of Borchani et al. [40] who stated that using this single-output method cannot obtain the most desirable predictive outcome, because it may neglect the dependencies between outputs. A possible explanation for this finding is that the dataset used in this study is not sufficiently large to provide enough inferences regarding correlations between selected features and these 8 outputs. Compared to the single-output ML models, the multi-output models require more information on how these features are correlated to the outputs, especially when this correlation is non-linear. Therefore, by further extending the dataset, it is possible to enhance the performance of the multi-output ML models. Furthermore, although each single-output model can provide better accuracy, it is still recommended to apply the multi-output ML models to predict all the outputs

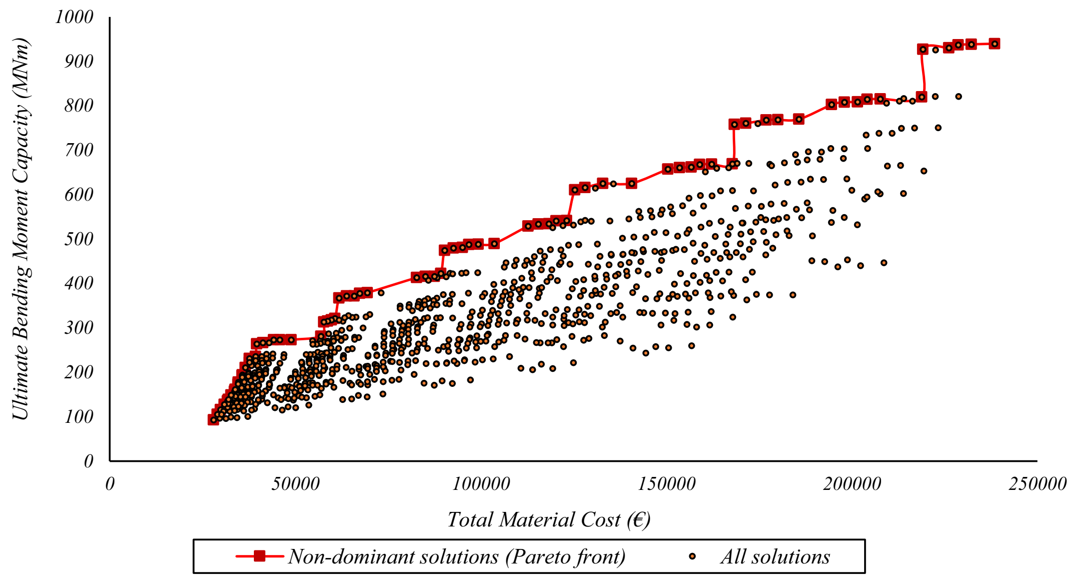

Fig. 12. The design solutions of the sub-dataset and the optimal solutions based on the WindBase dataset.

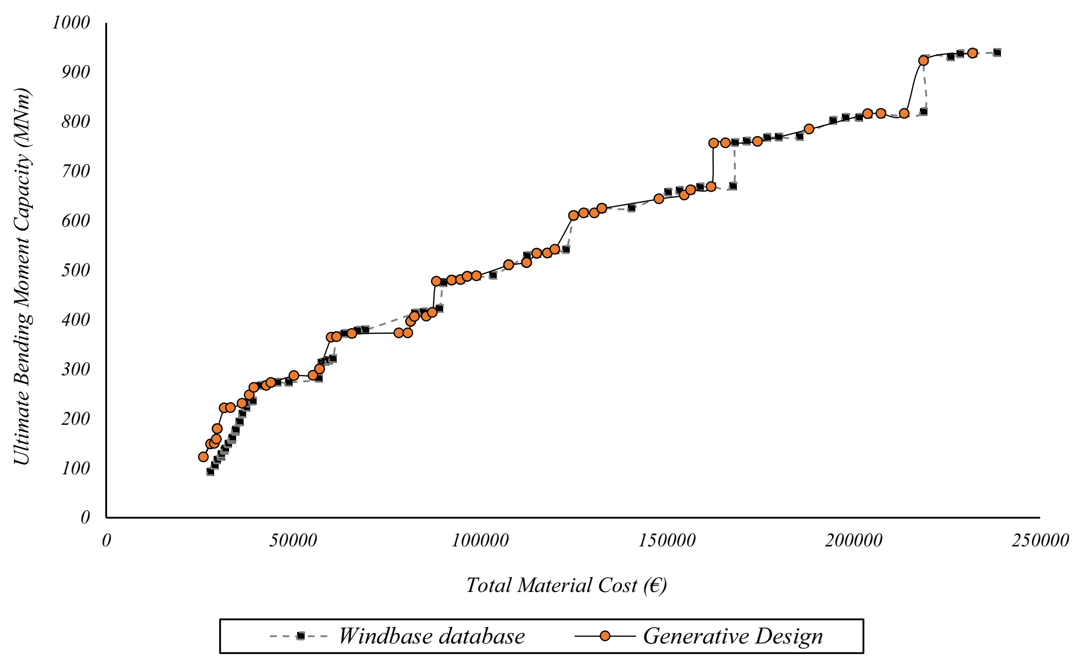

Fig. 13. The comparison between the optimal Pareto fronts obtained from the generative design and the sub-datase.

Table 12
The comparison of the hypervolume indicators and the total computational time for optimization.

Method	Hypervolume indicator	The total computational time for finding optimal solutions
Meta-model-based generative design method	0.58	29 mins
Traditional FEA-based design optimization method	0.57	38790 mins

simultaneously, given the fact that the total running time for using the stacked single-output models to predict 8 outputs simultaneously is more than the multi-output model, as shown in Table 9. Therefore, when extending these single-output models into broader applications such as generative design, the longer running time would considerably reduce the efficiency.

It is important to notice that the training of the ML model is a non-trivial task that can be time-consuming. However, it should be pointed out that once a model is developed based on a large enough dataset, the model can be easily reused for new projects and there is no need for continuous re-training. Given that companies already have a wealth of data about previously optimized designed foundations, these data can be leveraged to develop a model with good generalizability. Even if such a comprehensive dataset is not available at the moment, this research helps practicians see the value of structuring and maiming the historical data in a structured way for use in the same capacity proposed in this research.

Overall, the proposed method in this study can provide a significant time gain by reducing the computational time required from conducting the FEA in the static design optimization process of wind turbine foundations. Besides, developing the ML-based metamodels can also offer the designers a better understanding regarding the importance of each design variable and how a certain design variable influences the moment-rotation behavior of the wind turbine foundation. Last but not least, by applying this data-driven method in the wind turbine foundation design process, it will encourage the industry to establish a standard and consistent data structure as the basis for data mining, because this study provided new insight about how the usage of the historical data regarding the wind turbine foundation design can benefit the design optimization process.

6. Conclusion and future work

The study aimed to investigate how and to what extent the application of the metamodeling technique and the concept of generative design can facilitate the design process and optimization of the wind turbine foundation. To do that, a metamodel development process was proposed, mainly including the dataset establishment and the ML models development, and a generative design framework was proposed where the design optima can be obtained automatically under the given design constraints. In this study, two ML algorithms were selected, namely the multi-output RF and multi-output FFNN, and optimized using GA to determine the best model configuration, as well as the best combination of input features. In order to test and evaluate their predictive performance, a case study was conducted using the dataset provided by a Dutch design company, i.e., WindBase.

Based on the results presented in the case study, it can be concluded that metamodeling techniques have a great potential to be used as a complementary step to more accurate finite element modeling. It is also shown that the developed metamodel can be easily integrated into a design optimization method to constitute a generative design workflow that is both efficient and accurate.

In general, the multi-output RF model showed better performance with respect to the accuracy and computational time compared to single-output models. However, because, currently, only a limited number of design variables were considered in this research, further study is required to evaluate this method in a broader dataset. This would require a larger and more comprehensive database to store all variables that need to be considered as well as a consistent data structure. Besides, given the fact that the performance of using this GA-based method may be influenced by values of GA parameters, including the size of the population/offspring, the crossover rate, the mutation rate, and the stopping criteria, it is essential to determine the best configuration of the combination of these GA parameters. Potentially, this can be done by conducting a sensitivity analysis. Last but not least, currently, only the moment-rotation behavior of wind turbine foundations is considered as the output, it is also necessary to generate the proposed method to predict more outputs which can portray the more comprehensive structural performance of wind turbine foundations.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] F. Mey, M. Diesendorf, Who owns an energy transition? Strategic action fields and community wind energy in Denmark, Energy Res. Soc. Sci. 35 (2018) 108–117, https://doi.org/10.1016/j.erss.2017.10.044.

[2] S.A. Sherif, F. Barbir, T.N. Veziroglu, Wind energy and the hydrogen economy-review of the technology, Sol. Energy 78 (2005) 647–660, https://doi.org/10.1016/j.solener.2005.01.002.

[3] I. Komusanac, G. Brindley, D. Fraile, Wind energy in Europe in 2019, Brussels, Belgium. https://windeurope.org/wp-content/uploads/files/about-wind/statistics/WindEurope-Annual-Statistics-2019.pdf, 2020 (accessed September 7, 2020).

[4] K. Smith Stegen, M. Seel, The winds of change: how wind firms assess Germany's energy transition, Energy Policy 61 (2013) 1481–1489, https://doi.org/10.1016/j.enpol.2013.06.130.

[5] M. Bilgili, A. Yasar, E. Simsek, Offshore wind power development in Europe and its comparison with onshore counterpart, Renew. Sust. Energ. Rev. 15 (2011) 905–915, https://doi.org/10.1016/j.rser.2010.11.006.

[6] T. Burton, N. Jenkins, D. Sharpe, E. Bossanyi, Wind Energy Handbook, John Wiley & Sons, Ltd, Chichester, UK, 2011, https://doi.org/10.1002/9781119992714.

[7] M. Muskulus, S. Schafhirt, Design optimization of wind turbine support structures – a review, J. Ocean Wind Energy 1 (2014) 12–22. http://www.isope.org/publications (accessed September 7, 2020).

[8] J.C. Nicholson, Design of wind turbine tower and foundation systems: optimization approach, Univ. Iowa (2011), https://doi.org/10.17077/etd.bhnu76gr.

[9] M.B. Zaaijer, Foundation modelling to assess dynamic behaviour of offshore wind turbines, Appl. Ocean Res. 28 (2006) 45–57, https://doi.org/10.1016/j.apor.2006.03.004.

[10] J.S. Arora, Introduction to Optimum Design, Academic Press, 2012. http://dl.merc.ac.ir/handle/Hannan/3860 (accessed September 7, 2020).

[11] P.B. Loubser, A.R. Jacobs, Optimised design of wind turbine gravity foundations, in: Insights and Innovations in Structural Engineering, Mechanics and Computation: Proceedings of the 6th International Conference on Structural Engineering, Mechanics and Computation, 2016, pp. 5–7. https://www.abt.eu/bestanden/Afbeeldingen/Actueel/Publicaties/5109-1/Wind_turbine_foundations_Axel_Jacobs_Peter_Loubser.pdf (accessed September 7, 2020).

[12] S. Razavi, B.A. Tolson, D.H. Burn, S. Razavi, B.A. Tolson, D.H. Burn, Review of surrogate modeling in water resources, Water Resour. Res. 48 (2012) 7401, https://doi.org/10.1029/2011WR011527.

[13] N. Quirante, J. Javaloyes, J.A. Caballero, Rigorous design of distillation columns using surrogate models based on Kriging interpolation, AICHE J. 61 (2015) 2169–2187, https://doi.org/10.1002/aic.14798.

[14] A. Arisha, W. Abo-Hamad, Simulation optimisation methods in supply chain applications: a review, Ir. J. Manag. 30 (2010) 95–124, https://doi.org/10.21427/D7JJ6Z.

[15] A.O. Elfaki, S. Alatawi, E. Abushandi, Using intelligent techniques in construction project cost estimation: 10-year survey, Adv. Civil Eng. 2014 (2014), https://doi.org/10.1155/2014/107926.

[16] R.T. Rudolf, Reduced Order Simulation Surrogate for Wind Turbine Component Design, in: 11th World Congress on Structural and Multidisciplinary Optimisation, 2015, pp. 1–5. https://www.aeromech.usyd.edu.au/WCSMO2015/papers/1390_paper.pdf. (Accessed 16 December 2020).

[17] K. Maki, R. Sbragio, N. Vlahopoulos, System design of a wind turbine using a multi-level optimization approach, Renew. Energy 43 (2012) 101–110, https://doi.org/10.1016/j.renene.2011.11.027.

[18] B.A. Storti, J.J. Dorella, N.D. Roman, I. Peralta, A.E. Albanesi, Improving the efficiency of a Savonius wind turbine by designing a set of deflector plates with a metamodel-based optimization approach, Energy. 186 (2019), 115814, https://doi.org/10.1016/j.energy.2019.07.144.

[19] A. Albanesi, N. Roman, F. Bre, V. Fachinotti, A metamodel-based optimization approach to reduce the weight of composite laminated wind turbine blades, Compos. Struct. 194 (2018) 345–356, https://doi.org/10.1016/j.compstruct.2018.04.015.

[20] Y. Ju, C. Zhang, L. Ma, Artificial intelligence metamodel comparison and application to wind turbine airfoil uncertainty analysis, Adv. Mech. Eng. 8 (2016), https://doi.org/10.1177/1687814016647317, 168781401664731.

[21] A. Thiry, F. Bair, L. Buldgen, G. Raboni, P. Rigo, Optimization of Monopile Offshore Wind Structures, Advances in Marine Structures, 2011, pp. 633–642. http://www.vliz.be/projects/bencore/index.php?page=imis&module=ref&refid=296046&mid=9 (accessed September 7, 2020).

[22] R. Roy, A Primer on the Taguchi Method, Society of Manufacturing Engineers, 2010. ISBN 10: 087263468X.

[23] J. Spall, Introduction to Stochastic Search and Optimization: Estimation, Simulation and Control, Wiley & Sons, Inc., 2003. Online ISBN:9780471722137 | DOI:10.1002/0471722138.

[24] S.K. Dasari, A. Cheddad, P. Andersson, Random forest surrogate models to support design space exploration in aerospace use-case, in: IFIP Advances in Information and Communication Technology, Springer New York LLC, 2019, pp. 532–544, https://doi.org/10.1007/978-3-030-19823-7_45.

[25] H. Moayedi, M. Gör, M. Khari, L.K. Foong, M. Bahiraei, D.T. Bui, Hybridizing four wise neural-metaheuristic paradigms in predicting soil shear strength, Meas. J. Int. Meas. Confederation 156 (2020), 107576, https://doi.org/10.1016/j.measurement.2020.107576.

[26] H. Moayedi, M. Mosallanezhad, R. Nazir, Evaluation of Maintained Load Test (MLT) and Pile Driving Analyzer (PDA) in measuring bearing capacity of driven reinforced concrete piles, Soil Mech. Found. Eng. 54 (2017) 150–154, https://doi.org/10.1007/s11204-017-9449-1.

[27] W.T. Chan, Y.K. Chow, L.F. Liu, Neural network: an alternative to pile driving formulas, Comput. Geotech. 17 (1995) 135–156, https://doi.org/10.1016/0266-352X(95)93866-H.

[28] A. Ismail, D.S. Jeng, Modelling load-settlement behaviour of piles using high-order neural network (HON-PILE model), Eng. Appl. Artif. Intell. 24 (2011) 813–821, https://doi.org/10.1016/j.engappai.2011.02.008.

[29] P. Samui, T.G. Sitharam, Site characterization model using least-square support vector machine and relevance vector machine based on corrected SPT data (Nc),

Int. J. Numer. Anal. Methods Geomech. 34 (2010) 755–770, https://doi.org/10.1002/nag.837.
[30] A.J. Li, S. Khoo, A.V. Lyamin, Y. Wang, Rock slope stability analyses using extreme learning neural network and terminal steepest descent algorithm, Autom. Constr. 65 (2016) 42–50, https://doi.org/10.1016/j.autcon.2016.02.004.
[31] A. Wu, Z. Zeng, C. Fu, Data mining paradigm based on functional networks with applications in landslide prediction, in: Proceedings of the International Joint Conference on Neural Networks, Institute of Electrical and Electronics Engineers Inc., 2014, pp. 2826–2830, https://doi.org/10.1109/IJCNN.2014.6889362.
[32] K. Bakalis, M. Fragiadakis, D. Vamvatsikos, Surrogate modeling for the seismic performance assessment of liquid storage tanks, J. Struct. Eng. 143 (2017) 04016199, https://doi.org/10.1061/(ASCE)ST.1943-541X.0001667.
[33] M. Mardfekri, P. Gardoni, Probabilistic demand models and fragility estimates for offshore wind turbine support structures, Eng. Struct. 52 (2013) 478–487, https://doi.org/10.1016/j.engstruct.2013.03.016.
[34] J. Guo, Q. Wang, Y. Li, P. Liu, Façade defects classification from imbalanced dataset using meta learning-based convolutional neural network, Computer-Aided Civil Infrastruct. Eng. 35 (2020) 1403–1418, https://doi.org/10.1111/mice.12578.
[35] J. Chen, X. Gao, Y. Hu, Z. Zeng, Y. Liu, A meta-model-based optimization approach for fast and reliable calibration of building energy models, Energy 188 (2019), 116046, https://doi.org/10.1016/j.energy.2019.116046.
[36] H. Zheng, V. Moosavi, M. Akbarzadeh, Machine learning assisted evaluations in structural design and construction, Autom. Constr. 119 (2020), 103346, https://doi.org/10.1016/j.autcon.2020.103346.
[37] H. Kang, Y. Li, F. Wu, W. Guo, C. Huan, A system reliability analysis method for offshore wind turbine foundation, EJGE. 13 (2008) 1–10. http://ejge.com/2008/Ppr0848.pdf (accessed December 21, 2020).
[38] E. Alpaydin, Introduction to Machine Learning, MIT Press, London, UK, 2010. ISBN: 026201243X.
[39] S.B. Kotsiantis, Supervised machine learning: a review of classification techniques, Informatica 31 (2007) 249–268, https://doi.org/10.31449/inf.v31i3.148.
[40] H. Borchani, G. Varando, C. Bielza, P. Larrañaga, A survey on multi-output regression, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 5 (2015) 216–233, https://doi.org/10.1002/widm.1157.
[41] E. Spyromitros-Xioufis, G. Tsoumakas, W. Groves, I. Vlahavas, Multi-label classification methods for multi-target regression, ArXiv Preprint (2012) 1159–1168. ArXiv:1211.6581, http://rml.anu.edu.au/ (accessed September 7, 2020).
[42] J. Struyf, S. Džeroski, Constraint based induction of multi-objective regression trees, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer, Berlin, Heidelberg, 2006, pp. 222–233, https://doi.org/10.1007/11733492_13.
[43] D. Kocev, S. Džeroski, M.D. White, G.R. Newell, P. Griffioen, Using single- and multi-target regression trees and ensembles to model a compound index of vegetation condition, Ecol. Model. 220 (2009) 1159–1168, https://doi.org/10.1016/j.ecolmodel.2009.01.037.
[44] T.G. Dietterich, Ensemble methods in machine learning, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer Verlag, 2000, pp. 1–15, https://doi.org/10.1007/3-540-45014-9_1.
[45] D. Kocev, C. Vens, J. Struyf, S. Džeroski, Ensembles of multi-objective decision trees, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer Verlag, 2007, pp. 624–631, https://doi.org/10.1007/978-3-540-74958-5_61.
[46] J. Li, H. Hao, R. Wang, L. Li, Development and application of random forest technique for element level structural damage quantification, Struct. Control. Health Monit. (2020), https://doi.org/10.1002/stc.2678.
[47] D. Du, K. Li, M. Fei, A fast multi-output RBF neural network construction method, Neurocomputing. 73 (2010) 2196–2202, https://doi.org/10.1016/j.neucom.2010.01.014.
[48] N. An, W. Zhao, J. Wang, D. Shang, E. Zhao, Using multi-output feedforward neural network with empirical mode decomposition based signal filtering for electricity demand forecasting, Energy. 49 (2013) 279–288, https://doi.org/10.1016/j.energy.2012.10.035.
[49] V. Rouss, W. Charon, Multi-input and multi-output neural model of the mechanical nonlinear behaviour of a PEM fuel cell system, J. Power Sources 175 (2008) 1–17, https://doi.org/10.1016/j.jpowsour.2007.09.008.
[50] J. Zhao, M. Shi, G. Hu, X. Song, C. Zhang, D. Tao, W. Wu, A data-driven framework for tunnel geological-type prediction based on TBM operating data, IEEE Access. 7 (2019) 66703–66713, https://doi.org/10.1109/ACCESS.2019.2917756.
[51] J. Snoek, H. Larochelle, R.P. Adams, Practical Bayesian optimization of machine learning algorithms, in: Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012, NIPS 2012, Advances in Neural Information Processing Systems, 2012, pp. 2951–2959. https://collaborate.princeton.edu/en/publications/practical-bayesian-optimization-of-machine-learning-algorithms (accessed September 7, 2020).
[52] A.P. Engelbrecht, Fundamentals of Computational Swarm Intelligence, John Wiley & Sons Ltd, Chichester, England, 2005.
[53] Z. Arabasadi, R. Alizadehsani, M. Roshanzamir, H. Moosaei, A.A. Yarifard, Computer aided decision making for heart disease detection using hybrid neural network-genetic algorithm, Comput. Methods Prog. Biomed. 141 (2017) 19–26, https://doi.org/10.1016/j.cmpb.2017.01.004.
[54] A.L.I. Oliveira, P.L. Braga, R.M.F. Lima, M.L. Cornélio, GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation, Inf. Softw. Technol. 52 (2010) 1155–1166, https://doi.org/10.1016/j.infsof.2010.05.009.
[55] E. Matel, F. Vahdatikhaki, S. Hosseinyalamdary, T. Evers, H. Voordijk, An artificial neural network approach for cost estimation of engineering services, Int. J. Constr. Manag. (2019), https://doi.org/10.1080/15623599.2019.1692400.
[56] T.T. Wong, Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation, Pattern Recogn. 48 (2015) 2839–2846, https://doi.org/10.1016/j.patcog.2015.03.009.
[57] J. Ali, R. Khan, N. Ahmad, I. Maqsood, Random forests and decision trees, Int. J. Comput. Sci. Issues (IJCSI) 9 (2012) 272. www.IJCSI.org (accessed September 7, 2020).
[58] M.-H. Roy, D. Larocque, Robustness of random forests for regression, J. Nonparametric Stat. 24 (2012) 993–1006, https://doi.org/10.1080/10485252.2012.715161.
[59] M. Hayati, Y. Shirvany, Artificial neural network approach for short term load forecasting for Illam region, Int. J. Electr. Comput. Syst. Eng. 1 (2007) 121–125. https://www.researchgate.net/publication/239590608 (accessed September 7, 2020).
[60] S. Han, H. Kim, On the optimal size of candidate feature set in random forest, Appl. Sci. 2019 (9) (2019) 898, https://doi.org/10.3390/APP9050898.
[61] R. Genuer, J.-M. Poggi, C. Tuleau, Random forests: some methodological insights, ArXiv Preprint (2008). https://arxiv.org/abs/0811.3619v1 (accessed September 24, 2021), arXiv:0811.3619.
[62] I. Brown, C. Mues, An experimental comparison of classification algorithms for imbalanced credit scoring data sets, Expert Syst. Appl. 39 (2012) 3446–3453, https://doi.org/10.1016/J.ESWA.2011.09.033.
[63] X. Chen, H. Ishwaran, Random forests for genomic data analysis, Genomics. 99 (2012) 323–329, https://doi.org/10.1016/J.YGENO.2012.04.003.
[64] A.K. Langroodi, F. Vahdatikhaki, A. Doree, Activity recognition of construction equipment using fractional random forest, Autom. Constr. 122 (2021), 103465, https://doi.org/10.1016/J.AUTCON.2020.103465.
[65] Q. Wang, T.-T. Nguyen, J.Z. Huang, T.T. Nguyen, An efficient random forests algorithm for high dimensional data classification, ADAC 12 (2018) 953–972, https://doi.org/10.1007/S11634-018-0318-1.
[66] X. Gao, G.M. Lee, Moment-based rental prediction for bicycle-sharing transportation systems using a hybrid genetic algorithm and machine learning, Comput. Ind. Eng. 128 (2019) 60–69, https://doi.org/10.1016/j.cie.2018.12.023.
[67] T. Urbanek, Z. Prokopova, R. Silhavy, V. Vesela, Prediction accuracy measurements as a fitness function for software effort estimation, SpringerPlus. 4 (2015) 1–17, https://doi.org/10.1186/s40064-015-1555-9.
[68] A. Koumar, T. Tysmans, N. De Temmerman, R. Filomeno Coelho, L.A. Mira, Multi – criteria optimisation of a barrel vault structure for emergency relief, in: Proceedings of the IASS-SLTE 2014 Symposium “Shells, Membranes and Spatial Structures: Footprints”, International Association for Shell and Spatial Structures (IASS), 2014, pp. 1–8. https://www.researchgate.net/publication/280933800 (accessed November 9, 2020).
[69] L.I.W. Arnouts, T.J. Massart, N. De Temmerman, P.Z. Berke, Multi-objective optimisation of deployable bistable scissor structures, Autom. Constr. 114 (2020), 103154, https://doi.org/10.1016/j.autcon.2020.103154.
[70] S. Eleftheriadis, P. Duffour, B. Stephenson, D. Mumovic, Automated specification of steel reinforcement to support the optimisation of RC floors, Autom. Constr. 96 (2018) 366–377, https://doi.org/10.1016/j.autcon.2018.10.005.
[71] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer, Berlin, Heidelberg, Springer, 2000, pp. 849–858, https://doi.org/10.1007/3-540-45356-3_83.
[72] A. Konak, D.W. Coit, A.E. Smith, Multi-objective optimization using genetic algorithms: a tutorial, Reliab. Eng. Syst. Saf. 91 (2006) 992–1007, https://doi.org/10.1016/j.ress.2005.11.018.
[73] WindBase, (n.d.). https://windbase.eu/ (accessed December 23, 2020).
[74] T.W. Liao, P.J. Egbelu, B.R. Sarker, S.S. Leu, Metaheuristics for project and construction management – a state-of-the-art review, Autom. Constr. 20 (2011) 491–505, https://doi.org/10.1016/j.autcon.2010.12.006.
[75] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32, https://doi.org/10.1023/A:1010933404324.
[76] L. Breiman, Manual on Setting Up, Using, and Understanding Random Forests v3.1. http://oz.berkeley.edu/users/breiman/Using_random_forests_V3, 2002, 1. (accessed September 7, 2020).
[77] S. Salimi, M. Mawlana, A. Hammad, Performance analysis of simulation-based optimization of construction projects using High Performance Computing, Autom. Constr. 87 (2018) 158–172, https://doi.org/10.1016/j.autcon.2017.12.003.
[78] M. Kayri, I. Kayri, M.T. Gencoglu, The performance comparison of Multiple Linear Regression, Random Forest and Artificial Neural Network by using photovoltaic and atmospheric data, in: 2017 14th International Conference on Engineering of Modern Electric Systems, EMES 2017, Institute of Electrical and Electronics Engineers Inc, 2017, pp. 1–4, https://doi.org/10.1109/EMES.2017.7980368.
[79] L. Bradstreet, The Hypervolume indicator for Multi-Objective Optimisation: Calculation and Use, University of Western Australia, 2011. https://research-repository.uwa.edu.au/en/publications/the-hypervolume-indicator-for-multi-objective-optimisation-calcul (accessed December 21, 2020).
[80] S. Adusumilli, D. Bhatt, H. Wang, P. Bhattacharya, V. Devabhaktuni, A low-cost INS/GPS integration methodology based on random forest regression, Expert Syst. Appl. 40 (2013) 4653–4659, https://doi.org/10.1016/j.eswa.2013.02.002.
[81] T.D. Akinosho, L.O. Oyedele, M. Bilal, A.O. Ajayi, M.D. Delgado, O.O. Akinade, A. A. Ahmed, Deep learning in the construction industry: a review of present status

and future innovations, J. Build. Eng. 32 (2020), 101827, https://doi.org/10.1016/j.jobe.2020.101827.