MASTER THESIS

A Machine-Learning-Based Approach for the Design Optimization of Wind Turbine Foundations

AUTHOR
Qinshuo Shen
University of Twente
Construction Management and Engineering

EXAMINATION COMMITTEE
dr. J. T. Voordijk
dr. ir. F. Vahdatikhaki
dr. Ir. L. van der Meer
ir. T. van Dooren

DATE
12-08-2020

UNIVERSITY OF TWENTE.

Contents

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, particularly adopting renewable resources (Mey & Diesendorf, 2018). A typical case among these resources is wind energy, which is widely accepted as the fastest growing renewable energy source because of its availability, greenness, and cost-efficiency (Sherif, Barbir, & Veziroglu, 2005). 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 about wind energy boosted sharply. In 2019, approximately 15.4 GW of wind power capacity has been installed within European countries, making the total wind energy capacity in Europe rise to 205 GW, accounting for 15% of the total consumed electricity in Europe in 2019 (WindEurope, 2020). 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 (Stegen & Seel, 2013). 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, yet without compromising the structural performance.

Wind turbines can either be onshore or offshore, where the former has been utilized for a long time in the history, and usage of the latter is rather recent. Although winds are stronger and more stable in the offshore areas, onshore wind turbines still dominate the market, because offshore wind turbines are relatively more complex and costly to install and maintain (Bilgili, Yasar, & Simsek, 2011). Nearly 75% of the newly installed wind turbines in 2019 in Europe were onshore, making up 89% of the wind turbine capacity (WindEurope, 2020). The design and construction of onshore wind turbines may strike as straightforward and repetitive. However, the design process, particularly the foundation design, is complex and intricate comparing to other structures, due to the unsteady aerodynamic effects caused by the reaction between the turbulent wind and blade sections (Burton, Jenkins, Sharpe, & Bossanyi, 2011) (Muskulus & Schafhirt, 2014) (Oswell, Mitchell, Chalmers, & Mackinven, 2010) (Nicholson, 2011). Currently, static analysis is still commonly used for the design optimization of wind turbine foundations, which focuses on finding optimal solutions using non-linear analysis, under static load cases, constraints, and objectives (Muskulus & Schafhirt, 2014) (Arora, 2012). 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 (Loubser & Jacobs, 2016). 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 to evaluate each design alternative.

For the past few years, due to the expeditious expansion of the wind turbine network, a wealth of data has been generated from the designs of the wind turbine foundations, providing a rich database. Potentially, this database can be used to guide new designs towards optimality without going through a lengthy optimization process. However, this would require a meta-model that can correlate initial input to corresponding output and mathematically approximate the complex simulation model, to be conveniently used for rapid optimization. As a popular engineering method, meta-modeling can be connected with developing response surface surrogates to approximate the original simulation models using data-driven techniques (Razavi, Tolson, & Burn, 2012). When applying meta-modeling as the substitution of the original simulation models, the less intensive computation, reduced noisy output behavior, and the provision of gradients can be expected (Quirante, Javaloyes, & Caballero, 2015). Therefore, ideally, this meta-model can partly or fully replace the FEA process by mimicking the input-output behaviors in the original simulation models (Abo-Hamad & Arisha, 2011).

Due to the rapid development of computer technologies, Machine Learning (ML) approaches have shown the potential to tackle the aforementioned problem. A machine learning model is developed through a self-learning process where the training data is used to investigate and identify multi- and non- relationships and patterns between a set of input and the output data, to derive results that are more accurate and realistic (Elfaki, Alatawi, & Abushandi, 2014). Therefore, a significant time gain is expected by reducing the computational intensity caused by the current FEA simulation when the machine learning model can predict a close to optimal design more efficiently. Nevertheless, given the fact that conducting the FEA to assess the structural performance of wind turbine foundations may obtain several numbers of outcomes, to comprehensively portray these structural outcomes, it is essential to ensure the capability of the developed machine learning model in predicting multiple outputs. On this premise, this study focuses on developing the machine-learning-based mate-model to replace the FEA adopted in the optimization process of wind turbine foundations by solving a multi-output regression problem, in order to reduce the overall design time without compromising the accuracy. By conducting this research, it is expected to provide an opportunity to advance the understanding of how can the data-driven meta-modeling techniques be utilized as the substitution of complex simulation models, which are typically computationally expensive, in order to streamline the process of obtaining the optimal design of wind turbine foundations.

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 meta-model development, while section 4 demonstrates a case study and corresponding results, which can be used to validate the proposed method. The conclusion and future work are given in section 5.

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 (Nicholson, 2011). 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, which intend to reduce costs while safeguarding structural stability, many variables and constraints need to be considered (Muskulus & Schafhirt, 2014). However, the sheer number of pertinent parameters and variables significantly render the optimization process a demanding task. In addition, other unique characteristics of the wind turbine also challenge the design optimization of the wind turbine foundations, including the highly fluctuating and complex environment, fatigue damages, specialized analysis models and software, strong system inter-relation (Muskulus & Schafhirt, 2014).

A lot of experience has been accumulated in the research 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 provided important information on formulating the design optimization problem mathematically and rigorously to enable the assistance of computer-aided, automatics, and algorithmic approaches to derive the optimal solutions (Thiry, Bair, Buldgen, Raboni, & Rigo, 2011).

As reviewed by Muskulus and Schafhirt (2014), there are three major widely applied methods regarding the design optimization problems of wind turbine structures, namely the static analysis, frequency-domain analysis, and time-domain analysis. Among them, the static analysis focuses on finding optimal solutions under static loads and requires a large number of iterations in general cases. Typically, the objective of conducting this design optimization method is minimizing the material usage of the structure by revising parameters representing its geometry 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 (Muskulus & Schafhirt, 2014). Therefore, considering the potential computing time focuses should be paid on expediting the process while advancing the approximation techniques, improving reanalysis methods, and enhancing the efficiency of gradients evaluation. As is commonly used in offshore wind turbine structures design optimization, the frequency-domain analysis can provide good calculating performance in the fatigue assessment by linearizing irregular wave loads in the offshore engineering (Muskulus & Schafhirt, 2014) (Van Der Tempel, 2006). However, because this optimization method is rather confined to the area of offshore engineering, it is out of the scope of this research. Lastly, the time-domain analysis considers the integration of designs of different components of a wind turbine in different stages of the design process, given the fact that the wind turbine has a tightly coupled system (Vorpahl, Schwarze, Fischer, Seidel, & Jonkman, 2013). The biggest advantage of conducting this method is that it takes all non-linearities of the wind turbine operations into consideration, therefore, this method can ensure the highest accuracy (Van Der Tempel, 2006). However, because of the current simulation ratios of less than real-time, applying this optimization method is extremely time-consuming (Muskulus & Schafhirt, 2014).

2.2. Meta-model-based design optimization

Nowadays, the static analysis is still widely used in the design optimization of wind turbine structures, particularly in the design optimization of wind turbine foundations (Muskulus & Schafhirt, 2014) (Loubser & Jacobs, 2016). As previously mentioned, because the optimization process using this method is iterative, in order to reduce the time consumed, it is necessary to advance the approximation techniques, improve reanalysis methods, and enhance the efficiency of gradients evaluation. However, objective functions are not amenable to be efficiently evaluated using analysis methods such as FEA, especially when the structural response is too sensitive to changes in the geometry of the structure. Consequently, the inefficiency of FEA hinders the applicability of conducting the gradient assessment and invokes the employment of the simulation-based design optimization methods, including meta-model-based techniques, heuristic methods, and stochastic search (Gosavi, 2010) (Roy R. , 2010) (Spall, 2003).

Among them, meta-model-based (or surrogate-model-based) techniques show the potential to solve the prementioned problem regarding the inefficiency of the gradient assessment. Figure 1 represents how the meta-model 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 (Abo-Hamad & Arisha, 2011). The main objective of adopting the meta-modeling technique in design optimization is to solve complex optimization problems under the computational budget limitations. Because these techniques use the analytical approach to approximate the solution of the objective function rather than estimating the derivatives, they can provide an opportunity to explore the optimization space with greatly reduced computational time (Abo-Hamad & Arisha, 2011) (Dasari, Cheddad, & Andersson, 2019).

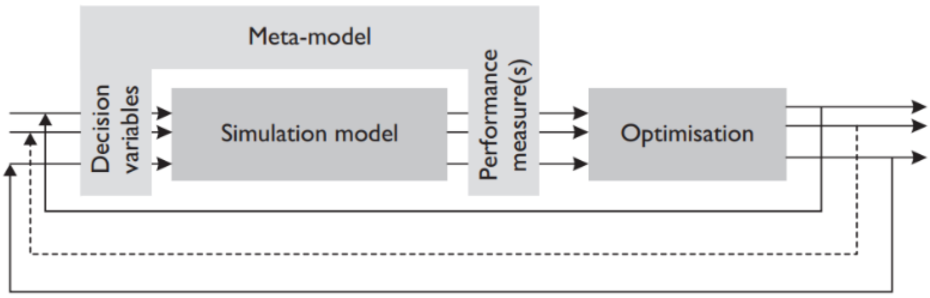

Figure 1. The general meta-model-based design optimization framework (Abo-Hamad & Arisha, 2011)

The meta-modeling can be generally categorized into two broad families, namely the response surface surrogates modeling and the lower fidelity modeling (Razavi, Tolson, & Burn, 2012). The former is concerned with applying data-driven techniques to approximate the correlations between explanatory variables that define the model response. In contrast to the response surface modeling, the lower fidelity modeling is a physically-based approximation technique, which can be regarded as the simplification of the original model and preserves the main body of the modeling process of the original simulation model. In general, the lower fidelity modeling can be expected to provide higher emulation to the unexplored input space and better performance in extrapolation. However, because the basis of the lower-fidelity modeling is the closeness between the lower-fidelity model and the original model, it often leads to the trade-off between the accuracy and computational cost (Razavi, Tolson, & Burn, 2012) (Jin, 2011).

2.2.1. Applications of meta-modeling in the AEC industry

Great attempts have been made to apply meta-modeling to the AEC industry over the past few decades to solve engineering problems. One of the research areas, where notable breakthroughs have been achieved in adopting the meta-modeling techniques, is geotechnical engineering. Given the fact that one of the major focuses of the geotechnical engineering is the study on the soil behavior, researchers have attempted to explore the potential of adopting meta-modeling techniques in the modeling of the soil behavior. Moayedi et al. (2020) represented a neural-metaheuristic-based meta-model to predict the soil shear strength, which shows the efficiency of modeling the non-linearities of the soil behavior and influential soil parameters. Besides, in terms of the area of the foundation design, previous studies have explored the potential of adopting neural-network-based meta-models in solving problems of the prediction of foundation capacity, especially the prediction of the pile capacity (Moayedi, Mosallanezhad, & Nazir, 2017) (Chan, Chow, & Liu, 1995) (Ismail & Jeng , 2011). In addition, numerous studies also indicated how can the meta-modeling techniques solve other geotechnical engineering problems with higher computational efficiency and better performance, including the subsurface exploration, slope stability assessment, and landslides assessment (Samui & Sitharam , 2010) (Li , Khoo , Lyamin , & Wang , 2016) (Wu , Zeng , & Fu, 2014).

Another fruitful area of adopting meta-modeling techniques in the AEC industry is the structural seismic performance assessment. Bakalis et al. (2017) demonstrated a low-fidelity 3-D 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. Similarly, Mardfekri and Gardoni (2013) also discussed a substitution to a refined 3-D model for the offshore wind turbine supporting structure using the probabilistic meta-model to reduce the sophistication of the original model and the computational cost and examine the seismic fragility. Apart from developing and utilizing low fidelity physically-based surrogate models to enhance the computational efficiency, investigations also have been made on applying the response surface surrogates in the seismic performance assessment. Ghosh et al. (2013) proposed several statistical-learning-based meta-modeling methods, including the benchmark meta-model using a polynomial regression model, emerging multivariate adaptive regression splines, neural networks, and support vector machines, to approximate the implicit relations between bridge design parameters and the bridge component seismic response. This research indicates that the significant computational efficiency and the desirable predictive performance regarding the seismic response of each component of the bridge can be expected by applying these response surface surrogates.

Furthermore, previous studies also show a wide application of the meta-model techniques in many other aspects of construction management. Guo et al. (2020) proposed a meta learning-based façade defects classification framework, using the collected image dataset and the concept of deep learning. In this study, the proposed meta learning-based model showed a strong capability to deal with the problem of an imbalanced image dataset that will be used as the basis of the Façade defects categorization. Besides, in the research area of the building engineering, Chen et al. (2019) also represented a meta-model which provides the potential to calibration performance of the building energy model, with reduced computational time and better reliability, using the Gaussian Process and Multiple Linear Regression to obtain a number of white-box simulation results. In addition, Zheng et al. (2020) proposed a machine-learning-based topological design approach to expedite the exploration process for shell structures. In this research, Zheng et al. utilized the neural network to develop a meta-model for rapid structural performance evaluation, which significantly streamlines the geometrical form-finding process. Besides, this study also reveals how can designers be provided with information on the effects of each design parameter on the final design outputs, by using the machine learning approach.

2.3. Machine learning methods

One of the most effective and widely applied approaches for developing the meta-model is using machine learning methods. Based on training data or past experience, inferences can be made using the theory of statistics within the machine learning model (Alpaydin, 2010). Therefore, one can expect that by learning the correlations between input variables and output obtained from FEA, the machine learning model can provide representation and an algorithmic solution for the inferences.

A typical division of machine learning algorithms can be made between supervised learning, unsupervised learning, and reinforcement learning, based on the usage of features in the training dataset (Kotsiantis, Zaharakis, & Pintelas, 2007). Supervised learning deals with instances with both input data and corresponding output data, while unsupervised learning is, by contrast, concerned with unlabelled datasets to investigate and characterize underlying structure (Kotsiantis, Zaharakis, & Pintelas, 2007) (Worden & Manson, 2007). Comparing to supervised and unsupervised learning, reinforcement learning receives information provided by the environment, and the agent will make decisions based on the performance of different actions (Alpaydin, 2010) (Kotsiantis, Zaharakis, & Pintelas, 2007). Furthermore, as stated by Cherkassky and Mulier (2007), problems addressed by supervised learning include classification and regression. The former requires the machine learning algorithm to determine the association of classes and a set or vector of measured quantities, while the latter refers to building a map between continuous input variables and continuous outputs.

2.3.1. Multi-output regression

The objective of the multi-output regression is simultaneously predicting multiple continuous-valued output variables. Borchani et al. (2015) 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, Spyromitros-Xioufis et al. (2012) 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 (Borchani, Varando, Bielza, & Larrañaga, 2015).

2.3.2. Multi-target Regression Trees

One widely used algorithm adaption method is Multi-target Regression Trees (MTRT), which enables the prediction of numerical multi-outputs simultaneously using regression trees (Struyf & Džeroski, 2006). A vector will be stocked in each leave within MTRT, which demonstrates a prediction for a single output variable (Kocev, Džeroski, White, Newell, & Griffioen, 2009). Normally, a top-down induction algorithm will be applied to build a multi-target regression tree. The top-down induction algorithm can be regarded as a recursive partitioning algorithm, which selects a test for the root node, and splits the training dataset into two subsets (in the case of the binary tree): one contains the records from the successful test, while the other contains records from failing test. Therefore, a recursive process will be developed. This recursive partitioning process will stop when the stopping criteria are satisfied, where the prediction vector is obtained and stocked in a leaf.

Besides, the ensemble method can also be used to build a set of models to combine predictions of each model and give predictions based on given training instances (Dietterich, 2000) (Kocev, Džeroski, White, Newell, & Griffioen, 2009). By ensembling a set of MTRTs, the prediction of multiple targets can be obtained by averaging the predictions of its models. This method can enhance the predictive performance of using a single MTRT (Kocev, Vens, Struyf, & Džeroski, 2007).

2.3.2.1. Random Forest

Random Forest (RF) is a tree-based machine learning algorithm, which ensembles a number of independent decision trees $\{h(\mathbf{x}, \Theta_k),\ k = 1,2, \dots\}$, in which every tree will be determined by the independently sampled random vector $\{\Theta_k\}$ (Breiman, Random Forests, 2001) (Dai, Gu, Zhao, & Qin, 2018). Figure 2 represents a typical structure of RF. By using the bootstrap sampling, the training set will be split into several subsets and trained in different randomized decision tree models, while the final prediction can be obtained by voting (Kocev, Džeroski, White, Newell, & Griffioen, 2009). For each node within the decision tree, a random size of the subset will be split from the training set and several input features will be randomly chosen from each subset as the input to the decision tree. Supposing the size of the subset is F, then its relation with the number of features M can be $F = 1, F = \sqrt{M}, F = \left(\log_2 M\right) + 1, F = \frac{M}{2}$, etc,. When $F = M$, RF will be equivalent to bagging algorithm, which is another ensemble method for MTRS (Kocev, Džeroski, White, Newell, & Griffioen, 2009).

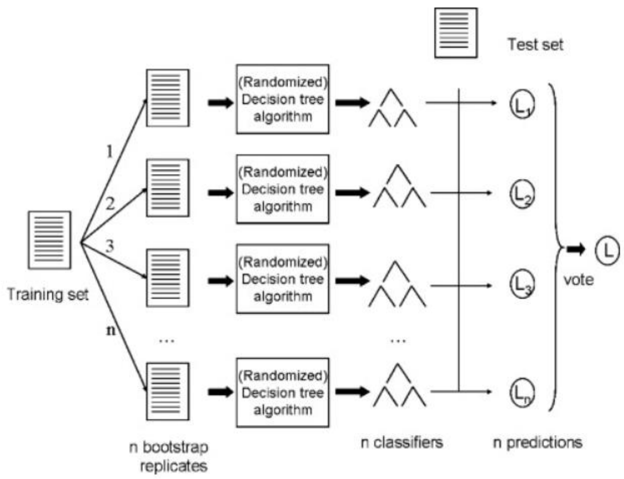

Figure 2. A typical RF structure (Struyf & Džeroski, 2006)

2.3.3. Feedforward Neural Network

Apart from RF, previous studies also reveal that the application of Artificial Neural Network (ANN) can provide potential to deal with the multi-output regression problems (Du, Li, & Fei, 2010) (An, Zhao, Wang, Shang, & Zhao, 2013) (Rouss & Charon, 2008).

As a widely adopted machine learning model, ANN is established by imitating the architecture of the human brain (Günaydin & Doğan , 2004). An ANN model can be built from the learning process towards the relationships between input and output provided from the training data, and recognize the non-linear relationships between the input and output in the hidden layer of the ANN architecture (Naik & Kumar, 2013).

Among all the commonly used ANN architectures, the feedforward neural network has been adopted broadly in previous studies. Typically, an FFNN consists of multiple nodes and layers (Arabasadi, Alizadehsani, Roshanzamir, Moosaei, & Yarifard, 2017). Nodes within the architecture of FFNNs can refer to neurons in the human brain, which will be grouped and interacted in each layer. Besides, FFNNs can contain a different number of layers. The first layer is called the input layer accepting features and input to the network, while the last layer refers to the output layer (Arabasadi, Alizadehsani, Roshanzamir, Moosaei, & Yarifard, 2017). In addition, there might be several hidden layers located between the input layer and the output layer, as shown in Figure 3.

Each node in one layer will be connected to all nodes in the previous layer with a different weight that determines the particular activation. The output of each neuron will thus be determined by the activation function f. A commonly used activation function is rectified linear unit ($ReLu$), where $f(x) = \max(0, x)$. The advantage of using $ReLu$ is that it can provide faster convergency speed and avoid suffering the vanishing gradient problems. In addition, when $f(x) = x$, the *linear* function is applied, which is usually used to build the output layer in the FFNN structure for regression problems. Furthermore, the weight which connects every node will be adjusted in every iteration based on the backpropagation algorithm, which will be executed in the forward phase to propagate signals to generate the prediction in the output layer, as well as the backward phase to adjust weights based on errors between the prediction and true values (Oliveira, Barga, Lima, & Cornélio, 2010).

When applying FFNN to solving the multi-output regression problems, An et al. (2013) have proposed a multi-input-multi-output strategy to use only one FFNN model to predict multiple outputs. Figure 3 illustrates the structure and prediction process using the FFNN model.

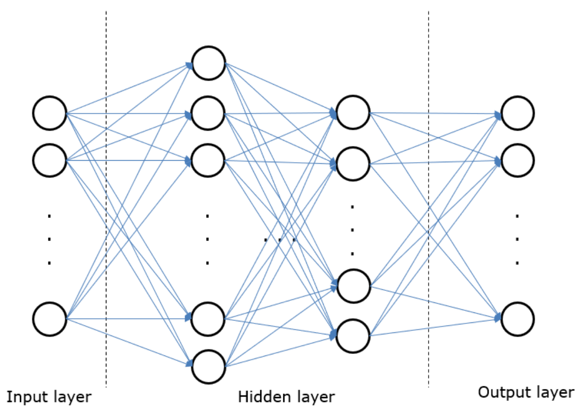

Figure 3. A typical FFNN structure

2.4. Genetic algorithm

The development of the machine learning model requires frequent tuning in terms of learning parameters and hyperparameters within the model until obtaining the best predictive performance (Snoek, Larochelle, & Adams, 2012). Among them, learning parameters are internal variables that represent the configuration of the model, whose values are estimated or trained from the data and normally will be adjusted automatically, such as weights in ANN. By contrast, model hyperparameters are external configurations of the machine learning model, whose values cannot be estimated and normally define higher concepts within the model, such as the learning rate in ANN or the number of estimators in RF. Therefore, it is essential to pay attention to find an effective and efficient method for hyperparameters optimization.

Corresponding studies have also shown how the application of heuristic 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 (Engelbrecht, 2005). 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 them, GA is the one of the most fruitful area in being integrated into the machine learning model to optimize the hyperparameters (Huang & Wang, 2006) (Oliveira, Barga, Lima, & Cornélio, 2010) (Bouktif, Fiaz, Ouni, & Serhani, 2018). Furthermore, optimization can also be implemented on the input variables involved in the machine learning model by eliminating irrelevant or redundant input variables. Olivera et al. (2010) also proposed a GA-based method for simultaneously optimize the selection of input features and optimize the setting of learning parameters and hyperparameters of three typical types of ML models, including the SVM, ANN, ad decision trees. The result of this research shows that the proposed approach can improve the performance of all machine learning models by reducing the amount of feature input set used. Therefore, in this study, a GA approach will be adopted to optimize the performance of machine learning models.

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 (Engelbrecht, 2006). 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 (Engelbrecht, 2006).

3. Proposed method

In this section, the process of the development of the proposed meta-model will be represented. Figure 4 shows an overview of the process, where the response surface surrogates modeling was applied. Overall, the process will be initiated by establishing a dataset with an array of design solutions of wind turbine foundations, which will be used as the input variables in the dataset, as well as corresponding structural outcomes obtained from conducting the FEA. Subsequently, the established dataset will provide the training set and the testing set, where the former will be used for the training process of several machine learning models and the latter will be used to validate each model. Besides, in the stage of training machine learning models, the GA-based method was applied for the feature selection and the optimization of hyperparameters to obtain the optimal machine learning model configuration. Lastly, the most favorable machine-learning-based meta-model will be chosen based on the predictive performance evaluated by the applied metrics. Details of this proposed developing process will be given in the following sections.

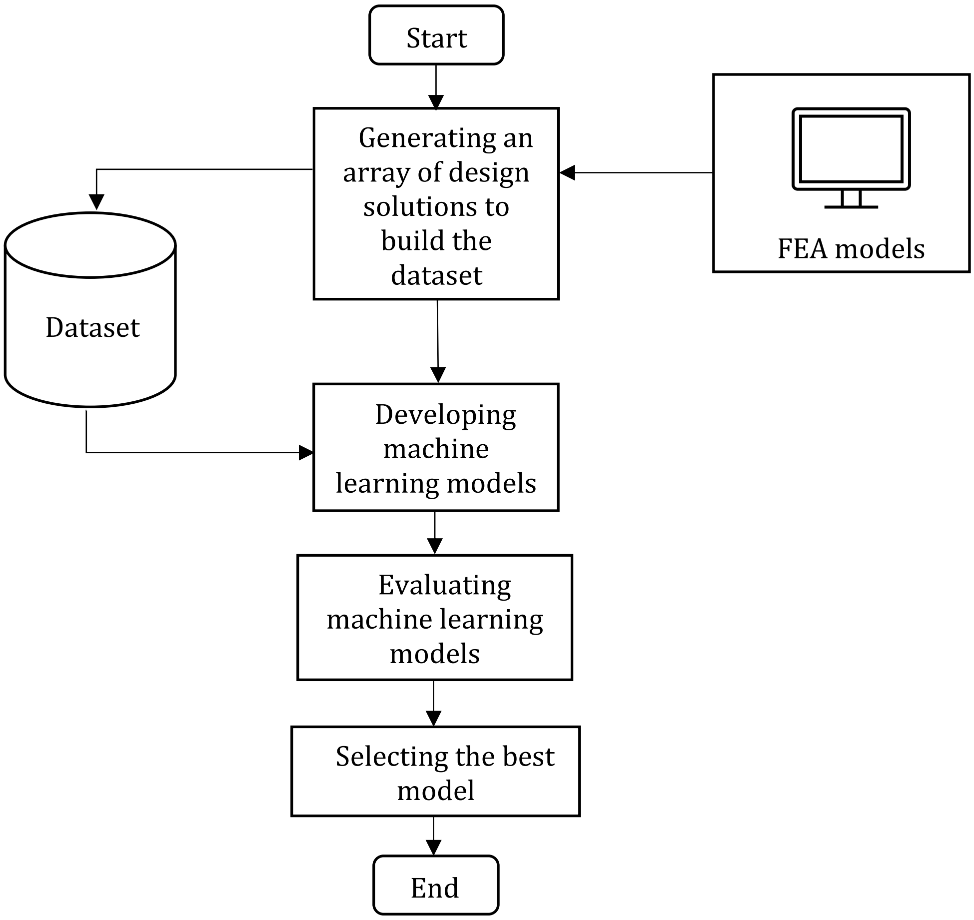

Figure 4. The framework of the meta-model development

3.1. The dataset establishment

The aim of this stage is to establish the dataset required for the machine learning model development. This dataset contains an array of design solutions, which will provide various design variables as input features in the developing phase of machine learning models and related structural performance of each design as outputs. Because in the actual design process, an extensive number of design variables will be involved, it is essential to verify and determine the most influential variables. Therefore, based on interviews held with several structural engineers with professional wind turbine foundations design experience and expertise in, and a similar study conducted by Nicholson (2011), several design variables were selected as shown in Table 1. In general, only the circular wind turbine foundation with steep slopes was taken into consideration in the proposed input data structure, given the fact that the circular shape is the most cost-efficient choice which can result in a considerable material saving without compromising the structural performance (Loubser & Jacobs, 2016) (Nicholson, 2011).

Therefore, several design variables that can be used to define the geometry of the circular wind turbine foundations were selected, including 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 represented by the 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.

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

Feature	Description
Base_diameter	The diameter of the base of the foundation.
Anchor_count	The number of anchors within the foundation.
Vertical_load_factor	The load factor which will be applied to calculating the vertical load transferred from the tower.
Base_slope_height	The height of the slope part of the foundation.
Base_side_height	The height of the foundation outer edge which is perpendicular to the base.
Top_rad_ratio	The reinforcement ratio of the radial reinforcement at the top part of the foundation.
Bottom_rad_ratio	The reinforcement ratio of the radial reinforcement at the bottom part of the foundation.
Top_tan_ratio	The reinforcement ratio of the tangential reinforcement in the top part of the foundation.
Bottom_tan_ratio	The reinforcement ratio of the tangential reinforcement in the top part of the foundation.

Besides, given the fact that the overturning moment is one of the most decisive load cases applied to the wind turbine foundation, it is essential to consider the moment capacity of the foundation as the main structural outcome (Muskulus & Schafhirt, 2014) (Nicholson, 2011). Therefore, the moment-rotation behavior of the wind turbine foundation was selected as the main structural outcome to be predicted, as well as the output included in the dataset, which can be obtained after conducting the FEA. Specifically, the moment-rotation behavior of the wind turbine foundation can be portrayed by the diagram shown in Figure 5.

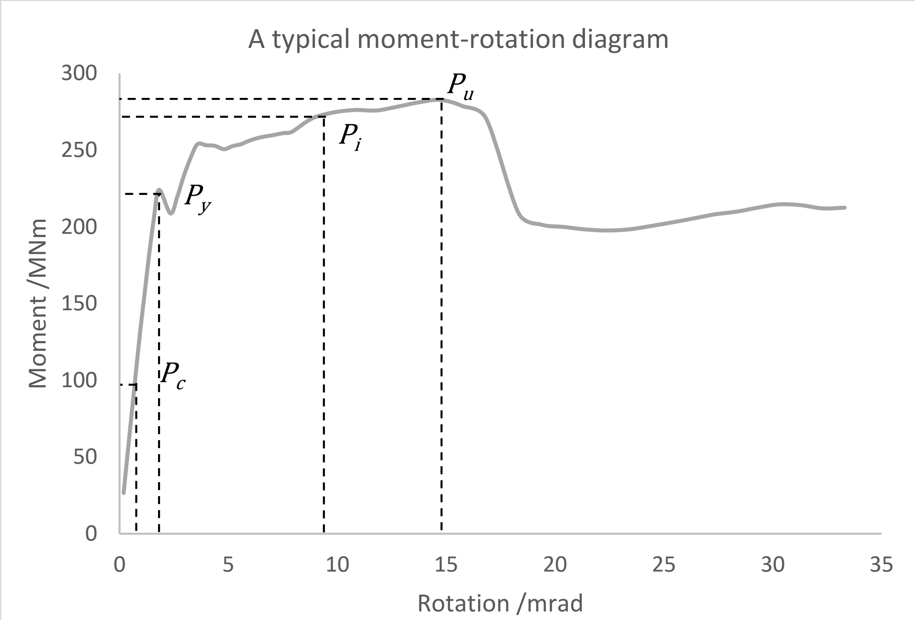

Figure 5. A typical moment-rotation diagram

The moment-rotation diagram shown in Figure 5 depicts the structural response of the wind turbine foundation as the increase in the rotation. The development of the curve will go through four main stages. 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, all the reinforcement groups within the wind turbine foundation will fail at P_u, where the foundation reaches its ultimate bending moment capacity.

Considering the difficulty in predicting the complete moment-rotation diagram using the machine learning method, it is essential to simplify the diagram using four key points that can portray the moment-rotation behavior of the foundation, as indicated in Table 2. These structural outcomes listed in Table 2 will be obtained using FEA by running simulations based on various design solutions.

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

Point in the diagram	Corresponding outputs	Description
P_c - initial cracking point	Initial cracking moment and rotation	The point where the reinforcement starts cracking.
P_y - initial yield point	Initial yield moment and rotation	The point where the reinforcement starts yielding.
P_i - initial failure point	Initial failure moment and rotation	The point where the reinforcement starts reaching the ultimate strain.
P_u - failure point	Failure moment and rotation	The point where all the reinforcement groups are failed.

3.2. Machine learning models development

In this stage of the meta-model developing process, algorithm adaption methods will be used as the main regression methods by adapting pattern recognition algorithms that can directly handle multi-output data. Machine learning 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 made in this study, as the performance benchmark. This can will be done by developing a machine learning model for each output in the dataset.

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

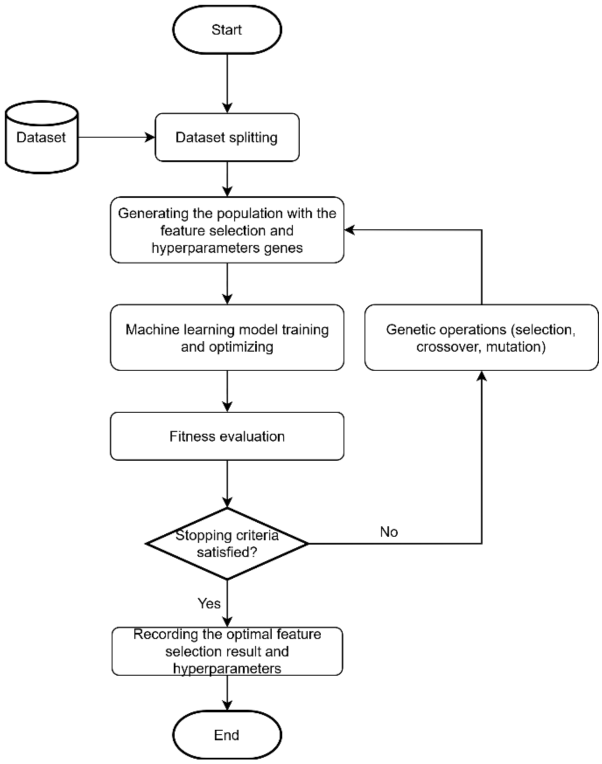

Figure 6. The framework of the proposed GA-based machine learning models developing process

Figure 6 provides an overview of the machine learning 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 be converted from genotypes into phenotypes.

(3) *Model training and the fitness evaluation*: both the training subset and the testing subset will be modified based on the feature selection results to train and validate machine learning models, and the fitness of each individual will be calculated.

(4) *Termination criteria assessment*: 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.

(5) *Genetic operations*: the new population will be generated after the crossover and mutation.

3.2.1. K-fold cross-validation

Typically, in order to ensure that the machine learning model after training can obtain a desirable generalization capability, the dataset will be split into the train set, the validation set, and the test set. The train set will be used for training the model, the validation set will be used to evaluate the model obtained after the training process, and if the evaluation is successful, the test will be eventually made on the test set to access the generalization ability of the model. However, this traditional dataset splitting method is sensitive to the proportions of three sets. Besides, the distribution of data within these three sets will also change, which may influence the performance of the model obtained. Last but not least, by splitting the original dataset into three sets, the number of data involved in the train set will decrease, which will influence the training process.

One widely adopted solution to address this problem is using the k-fold cross-validation to ensure the generalization ability of the machine learning model and prevent the overfitting problem, without making the model being sensitive to the data distribution. Using this strategy, the original dataset will be randomly divided into k disjoint folds with the approximately same size. Among them, $k - 1$ folds will be used as the training set to enable the self-learning process of the machine learning model, while the remaining fold will be used as the test set (Wong, 2015). Then, by averaging performance metrics obtained in the process, the final performance of the model can be calculated.

Normally, the value of k can be 10 or 5, depending on the size of the original dataset, where the higher the value of k is, the longer the computational time will require. Jung (2018) also provided an empirical approach to determine the value of k, by setting $k \approx \log N$ and ensure that $\frac{N}{k} > 3d$, where N stands for the size of the original dataset, and d stands for the number of input features. In this article, the value of k will be determined using this empirical method.

3.2.2. The GA-based model optimization approach

The optimization of machine learning 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 machine learning algorithm. Subsequently, genotypes of each chromosome will be converted to phenotypes, such as the information whether a certain feature will be 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 generate new solutions 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.

3.2.2.1. Chromosome design

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 machine learning algorithm. Therefore, the chromosome will be divided into two parts as shown in Figure 7, 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.

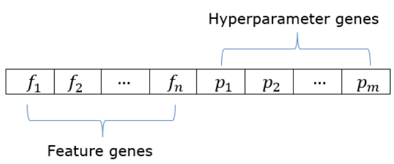

Figure 7. The gene combination

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 machine learning algorithms were selected, namely RF and FFNN. From previous studies, these two machine learning algorithms have shown the potential to deal with multi-output regression problems, with good predictive performance, computational speed, scalability, ease of use, and extensibility (An, Zhao, Wang, Shang, & Zhao, 2013) (Borchani, Varando, Bielza, & Larrañaga, 2015). Specifically, RF can provide good performance on working with large datasets, and it has the ability to overcome the overfitting problem and outliers (Ali, Khan, Ahmad, & Maqsood, 2012) (Roy & Larocque, 2012). 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 (Hayati & Shirvany, 2007).

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.

Table 3. Explanation of hyperparameters required to be optimized

Machine learning algorithm	Hyperparameter	Description
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 from the train set that is required to split an internal node.
	min_samples_leaf	The minimum number of samples required to be at a leaf node.
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.

3.2.2.2. Fitness function

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 (Gao & Lee, 2019). In this article, because the purpose is using a GA-based method to find the optimized machine learning model configuration, no specific value for the fitness to be achieved will be given. Therefore, the fitness function will only be used for the selection stage of the process. Instead, the fitness function will directly reflect the predictive performance of the machine learning model.

Several performance evaluation metrics have been introduced by Borchani et al. (2015) for multi-output regression problems. In this study, the mean squared error (MSE) will be applied in the fitness function and also as the metric. According to Urbanek et al. (2015), 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 Equation 1, according to Borchani et al. (2015).

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

where d stands for the number of outputs, N_{test} stands for the size of the test dataset, $y_i^{(l)}$ stands for the actual value of the ith output corresponding to $X^{(l)}$ in the test dataset, and $\hat{y}_i^{(l)}$ represents the predicted value of the ith 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 model training phase. Therefore, for every output, the range of potentially calculated MSE will between 0 to 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.

3.2.2.3. *Stopping criteria*

Stopping criteria will be used as the mechanism to determine when the optimization process can be finished. As indicated in Equation 2, the convergence criterion indicates the change in the value of the fitness objective function in the last ten iterations (Querin, Victoria, Gordoa, Ansola, & Martí, 2017).

$$\varepsilon_i = \frac{\sum_{i-9}^{i-5} fitness - \sum_{i-4}^{i} fitness}{\sum_{i-4}^{i} fitness} \quad (2)$$

where i is the current number of the iteration, which should be higher than 10.

Typically, the convergence limit (ε_{lim}) should be between 0.001 and 0.01 (Querin, Victoria, Gordoa, Ansola, & Martí, 2017). In this study, the value of ε_{lim} will be set to 0.01. Therefore, the optimization process will end after 10 iterations if ε_i is less than ε_{lim}.

4. Case study

4.1. Dataset

In this research, a case study was conducted to test the proposed method. The dataset used in this research was provided by WindBase, which is a department within the Dutch design company ABT with over 25 years of experience in the wind turbine foundation designs, and over 2800 wind turbine foundations have been designed and built. In order to ensure the consistency of the data structure, this dataset only contains design variables determined and described in section 3, where Table 4 indicated the range of each variable. Besides, corresponding outputs were obtained by running calculations using FEA with different design solutions.

Table 4. Input features from the dataset

Feature	Range
Base_diameter	From12.5 m to 25 m with a step of 2.5m
Anchor_count	From 60 to 120 with a step of 30
Vertical_load_factor	0.9 and 1.35
Base_slope_height	From 0 to 2.3125 m
Base_side_height	From 1.5 m to 2.1875 m
Top_rad_ratio	From 0.041% to 0.63%
Bottom_rad_ratio	From 0.041% to 0.63%
Top_tan_ratio	From 0.047% to 0.36%
Bottom_tan_ratio	From 0.09% to 0.71%

4.2. Parameters in the GA process

Parameters for the GA process have been pre-defined, as indicated in Table 5.

In this case study, the size of the initial population will be set to 50, which means 50 individuals will be included as the input of the GA process. Besides, during the crossover stage, the crossover rate will be set to 0.8, which represents the possibility of two randomly selected individuals after the selection to exchange gene fragments. Furthermore, the possibility of the mutation occurring in each offspring generated after the crossover phase will be 0.1.

Table 5. Pre-defined GA parameters

GA parameter	Description	Value
Population size	The number of individuals that will be evaluated and selected.	50
Offspring size	The number of individuals which will be re-generated after one iteration of the proposed GA process.	50
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

4.3. Results

As described in section 3.2.2.2, MSE was selected to be 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 will also be used as the main metric to indicate the model predictive performance. However, MSE cannot intuitively indicate how accurate the developed machine learning model can be. Therefore, another performance metric named PRED(25) will also be applied in this research, which represents the percentage of predictions falling within 25% of the true value (Oliveira, Barga, Lima, & Cornélio, 2010). Therefore, the overall predictive performance of multi-output models will be determined by MSE and PRED(25). The former can be calculated by summing up the MSE obtained on each output as indicated in Equation 1, while the latter will be the average of the PRED(25) achieved on each output. Unlike MSE, higher value achieved in PRED(25) means higher accuracy of the prediction.

Table 6 indicates the result after testing GA-based machine learning models using this dataset. Furthermore, in order to provide a more intuitive representation of results, regression plots for each multi-output model were also provided, as shown in Figure 8 and Figure 9 respectively. It is worth noting that for the multi-output RF and the multioutput-FFNN, not only the overall predictive performance will be given, but also the performance on each output. Besides, for every sing output, two single-output machine learning 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 was developed to predict one specific output independently.

Table 6. Results obtained from the case study

Output		Multi-output RF	Single-output RFs	Multi-output FFNN	Single-output FFNNS
Overall	MSE	0.0169	-	0.0167	-
	PRED(25)	83.08%	-	75.81%	-
Initial cracking rotation	MSE	0.0018	0.0018	0.0013	0.0018
	PRED(25)	77.57%	93.64%	63.89%	87.57%
Initial cracking moment	MSE	0.0013	0.0013	0.0014	0.0013
	PRED(25)	79.76%	95.07%	70.89%	91.67%
Initial yield rotation	MSE	0.0025	0.0024	0.0026	0.0026
	PRED(25)	82.18%	92.90%	75.38%	84.75%
Initial yield moment	MSE	0.0014	0.0014	0.0014	0.0016
	PRED(25)	87.87%	96.52%	87.09%	94.03%
Initial failure rotation	MSE	0.0010	0.0006	0.0007	0.0010
	PRED(25)	73.39%	95.82%	60.17%	86.65%
Initial failure moment	MSE	0.0014	0.0012	0.0013	0.0014
	PRED(25)	77.49%	96.84%	63.59%	96.38%
Failure rotation	MSE	0.0062	0.0062	0.0064	0.0100
	PRED(25)	89.46%	91.41%	88.35%	87.25%
Failure moment	MSE	0.0013	0.0010	0.0011	0.0012
	PRED(25)	96.96%	97.81%	97.12%	97.51%

Table 7 represents the configuration of obtained optimal multi-output models using these two machine learning algorithms. In terms of the feature selection, the two developed models applied different feature combinations. For the multi-output RF model, the feature *Top_rad_ratio* was reduced, while there is no feature has been reduced by using the multi-output FFNN model.

Table 7. Results of the feature selection and optimal model configurations

Machine learning algorithm	Reduced feature	Hyperparameter	Value
RF	*Base_slope_height*	n_estimators	856
		max_depth	86
		min_samples_leaf	2
		min_samples_split	3
FFNN	None	n_layers	3
		n_nodes	26-17-31
		epochs	500
		learning_rate	0.002
		batch_size	10

The structure of the value of hyperparameter n_nodes means there are 12 nodes, 24 nodes, and 11 nodes in the first, the second, and the third hidden layer respectively.

From the result, it is clear that using multi-output RF can provide closed predictive performance regarding the calculated MSE. However, because the calculated PRED(25) of the multi-output RF is considerably higher than the multi-output FFNN, the former can be regarded as the more desirable choice that can provide better predictive performance in terms of predicting all eight outputs. Besides, in general, while the calculated PRED(25) is higher than 75%, the developed regression model can be regarded as acceptably accurate (Kumar, Ravi, Carr, & Kiran, 2008). Therefore, based on these results, using the multi-output RF model will have poor performance in predicting the initial failure rotation, while the multi-output FFNN fails to predict the initial failure rotation, the initial failure moment, the initial cracking rotation, and the initial cracking moment on an acceptable level.

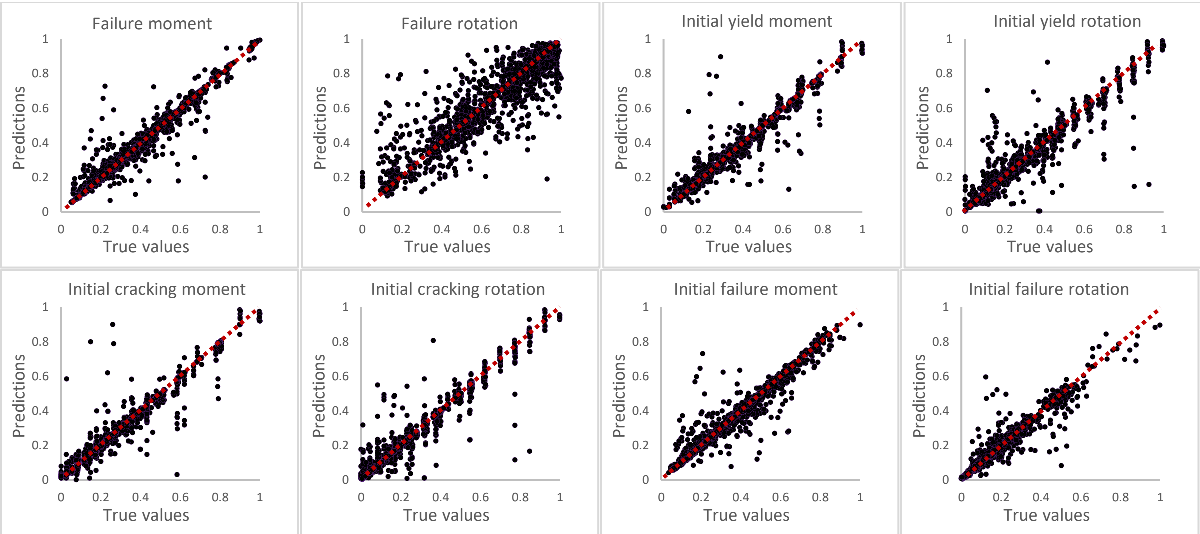

Figure 8. Regression plots of the multi-output RF model

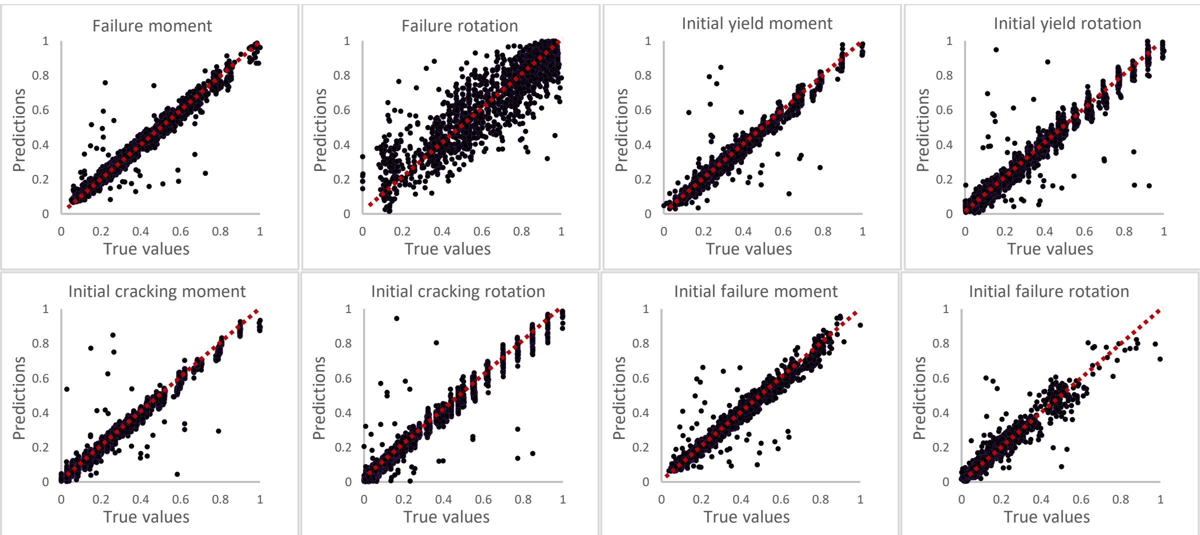

Figure 9. Regression plots of the multi-output FFNN model

Furthermore, Figure 10 and Figure 11 indicate the regression plots of each output predicted by corresponding single-output models. When comparing the multi-output models with corresponding single-output models, the comparison shows that the usage of the latter can generally provide improved predictive performance, except the prediction on the failure rotation. This finding reveals that it is difficult to use the proposed GA-based optimization approach to find the optimal solution regarding the feature selection and hyperparameters configuration to obtain accurate predictions on every output simultaneously. Therefore, for the prediction on a single output, it is recommended to adopt the single-output machine learning models, where using single-output RF models can generally provide better predictive performance on each output. However, when it comes to the prediction of these 8 outputs simultaneously, considering the time consumed for developing an array of single-output machine learning models to predict all the outputs simultaneously is dramatically longer than only developing one multi-output model, as shown in Table 8, the latter should be regarded as the most desirable choice. Besides, considering the overall accuracy and the computational time required for the model development, the multi-output RF model is recommended to be utilized to function as the meta-model as the substitution of the FEA.

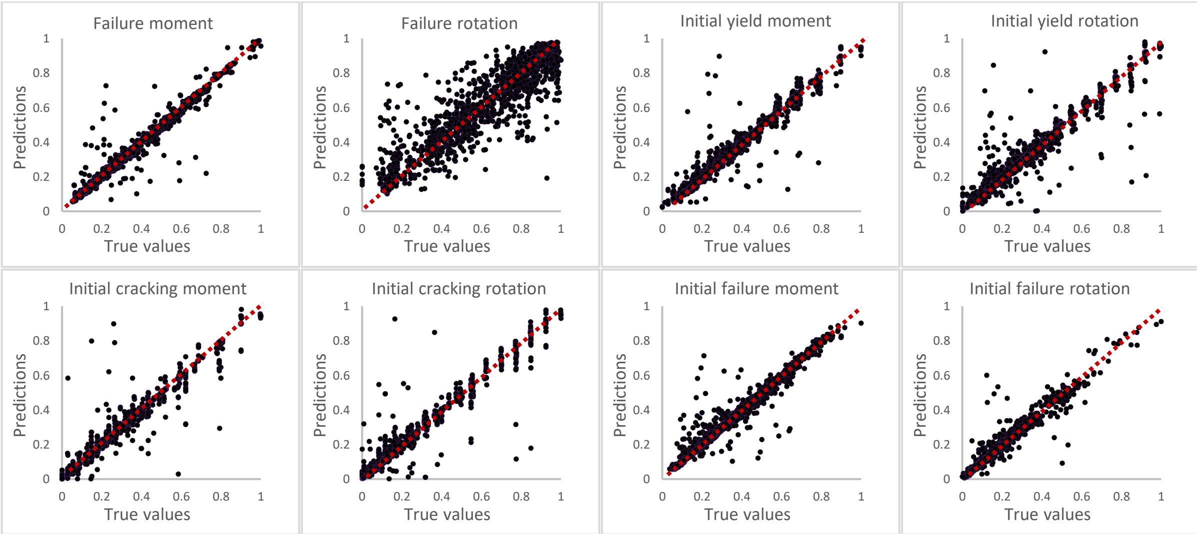

Figure 10. Regression plots of the single-output RF models

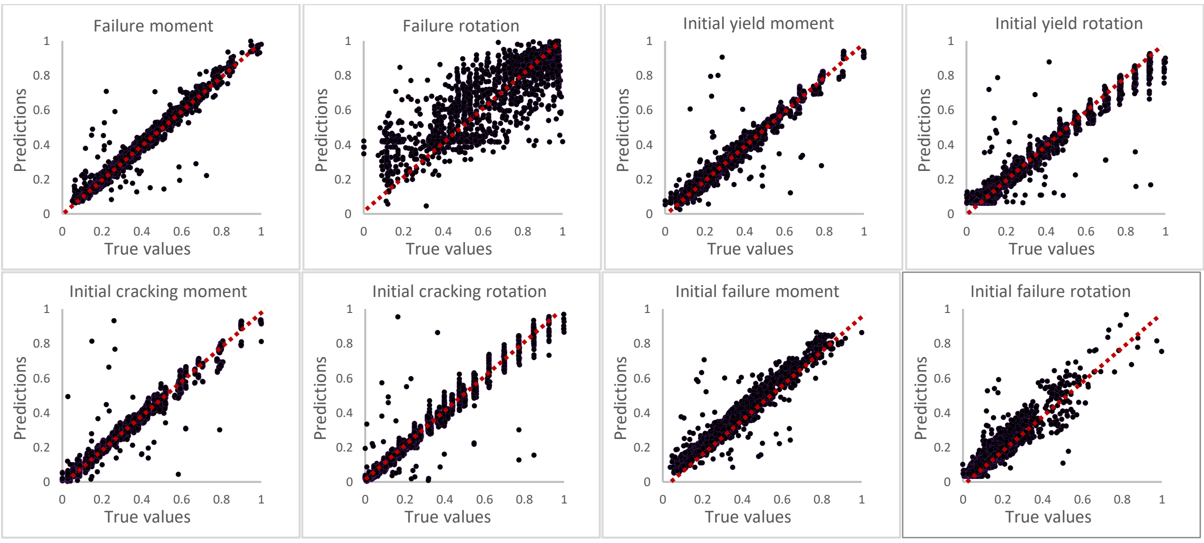

Figure 11. Regression plots of the single-output FFNN models

Table 8. The average computational time for the development of each machine learning model

Model	Multi-output RF	Single-output RFs	Multi-output FFNN	Single-output FFNNS
Average computational time	200 min	183 min	455min	415 min

4.4. Model interpretability

After developing machine learning models in this case study, it is 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 machine learning models are normally regarded as "black box" tools.

4.4.1. Sensitivity analysis

One of the most effective methods to investigate the interpretability of models is conducting the sensitivity analysis to examine how the influence of each input feature can affect the predictive performance of models. In this study, the analysis will be done by using permutations to randomly shuffle a single feature value, in which the feature importance can be obtained based on the change in the model score (Breiman, 2001). It is worth noting that the feature importance obtained using permutation cannot reflect the inherent predictive value of features, but 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 (Breiman, 2002). Therefore, the relative impurity-based feature importance was also be calculated as the comparison.

Because in the previous section, results show that multi-output RF is the most favorable choice with higher predictive performance and less computational time, only the multi-output RF model gained from the case study using the GA process was used to conduct this sensitivity analysis. It is worth noting that the selected model may reduce several features during the development phase due to the feature selection of the GA process.

Besides, based on the results of the case study represented in section 4.3, the predictive performance on the initial failure rotation is not on an acceptable level using the obtained optimal multi-output RF model, while the prediction on the failure rotation cannot be improved by using the corresponding single-output RF model either. Therefore, one can conclude that these two outputs can be regarded as the most difficult outputs to be predicted. In order to further investigate how features contribute to the prediction of these two outputs, extra sensitivity analyses were made on obtained optimal single-output RF models that predict the failure rotation and the initial rotation respectively.

4.4.2. Results of the model interpretation

Figure 12 shows the results of this permutation-based sensitivity analysis on the optimal multi-output RF model. In total, eight features have been selected in the optimized multi-output models after the GA-based feature selection process. The feature *Base_slope_height* has been reduced from the model training phase.

In addition, the impurity-based feature importance is indicated in Figure 13. The results of the impurity-based feature importance match the results obtained from the sensitivity analysis. Therefore, for this optimized multi-output RF model, *Base_diameter, Base_side_height*, and *Bottom_tan_ratio* are the most important features that will be more decisive in influencing the moment-rotation behavior of wind turbine foundations, while *Anchor_counts* has the least importance.

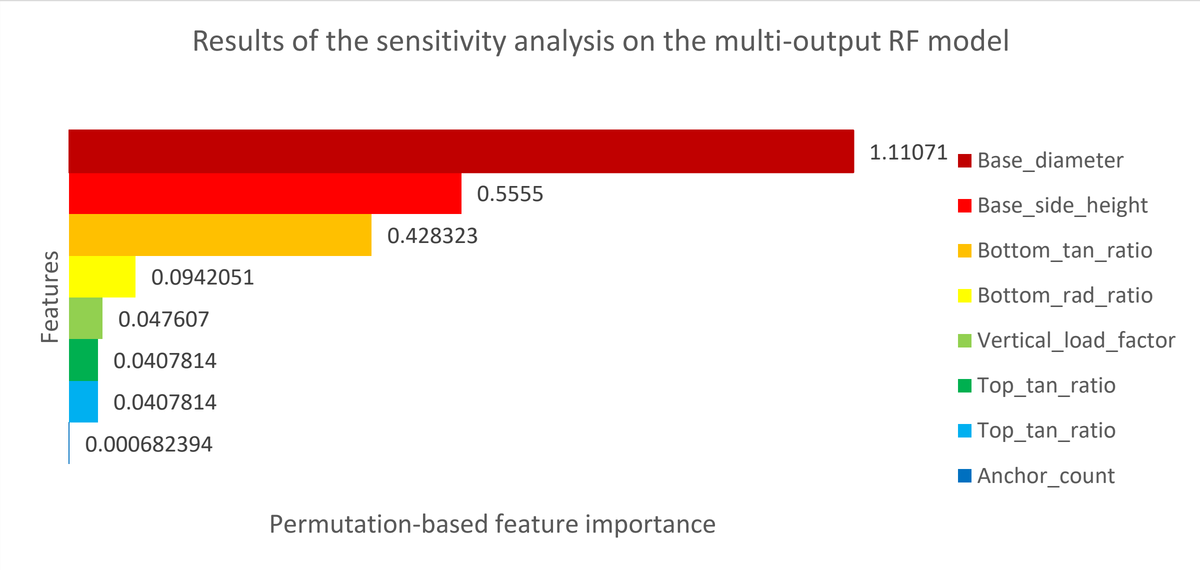

Figure 12. The feature importance obtained from the sensitivity analysis on the multi-output RF model

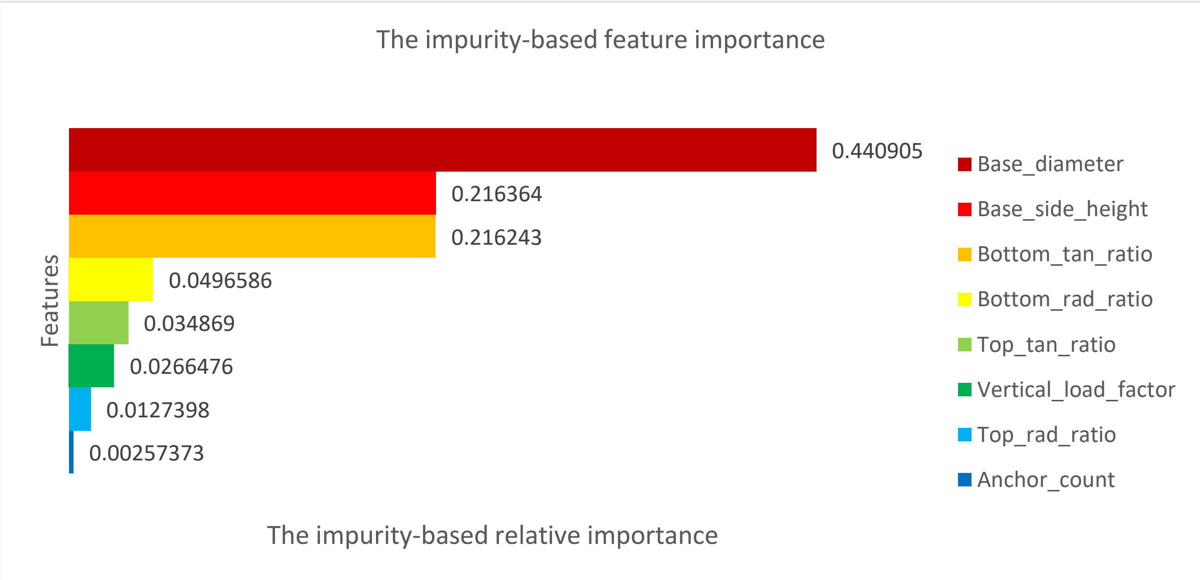

Figure 13. The relative impurity-based feature importance on the multi-output RF model

Besides, Figure 14 and Figure 15 show the permutation-based feature importance of two single-output RF models, which predict the failure rotation and the initial failure rotation respectively.

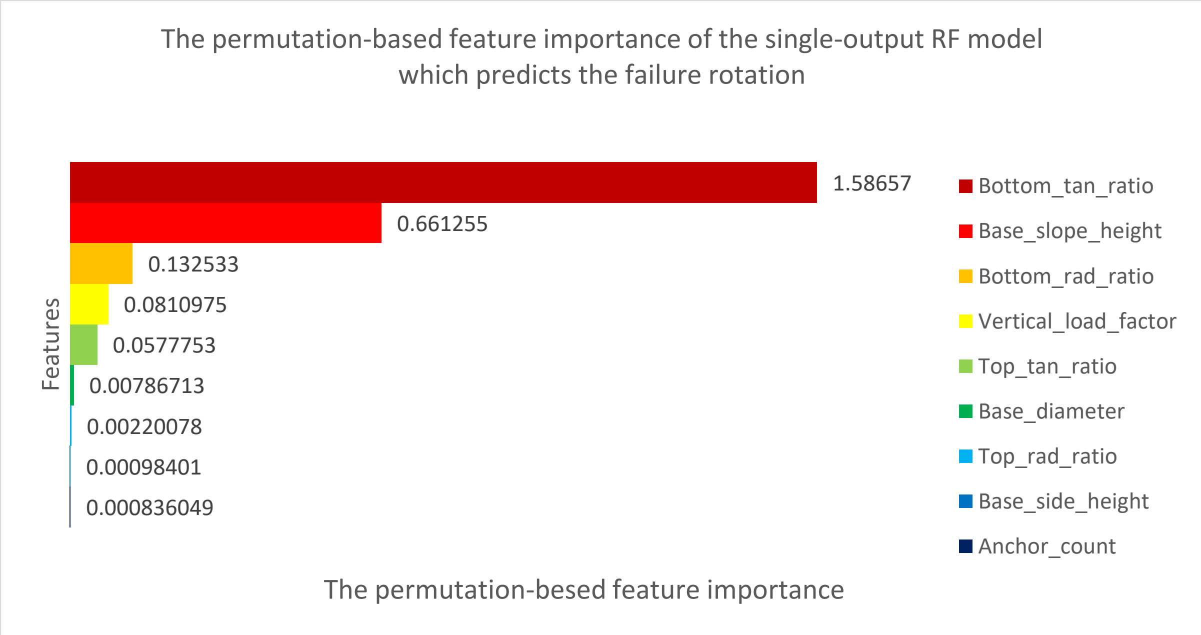

Figure 14. The permutation-based feature importance of the single-output RF model which predicts the failure rotation

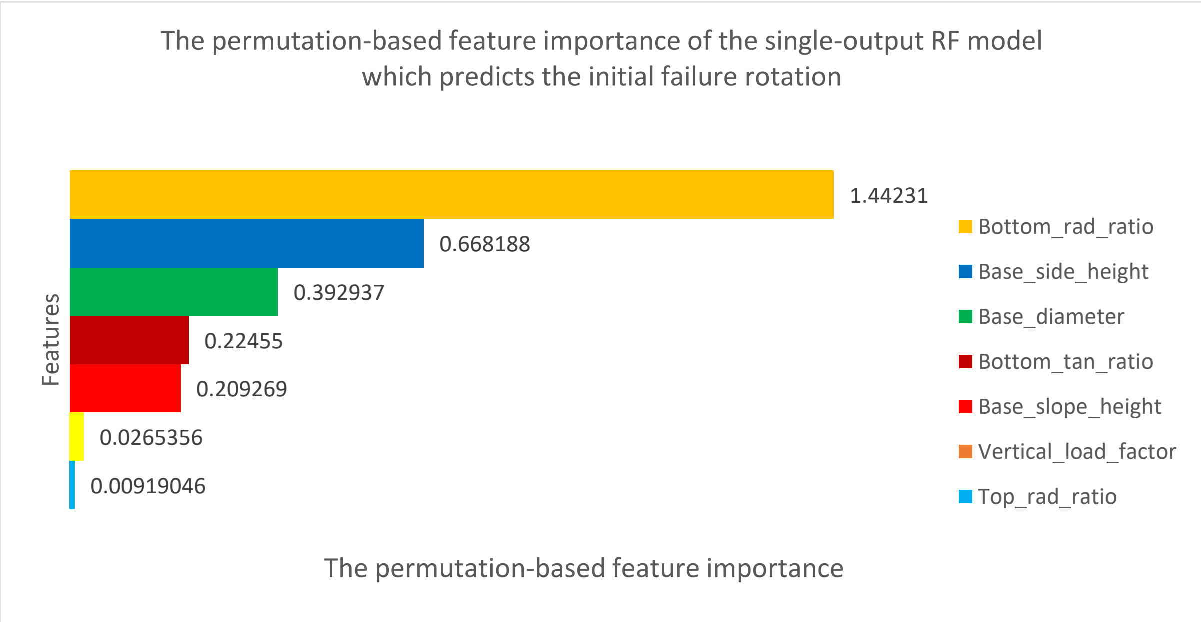

Figure 15. The permutation-based feature importance of the single-output RF model which predicts the initial failure rotation

Based on the result provided in section 4.3, neither the multi-output RF model nor the corresponding single-output RF model can predict the failure rotation very accurately. From the result shown in Figure 14, in contrast to the multi-output RF model, the feature *Base_slope_height* was not removed from the feature selection process of the single-output RF model for predicting the failure rotation. Instead, this feature has a considerable effect on the failure rotation, which is not in line with the result indicated in Figure 12 and Figure 13. Therefore, the multi-output RF cannot make full use of the inferences provided by the decisive features shown in Figure 14, which leads to the poor predictive performance on the failure rotation. Furthermore, another possible reason could be the fact that the size of the currently used dataset is not sufficiently large. Consequently, neither the multi-output RF model nor the single-output RF model can realize a better performance prediction on the failure rotation because the current dataset cannot provide enough information about how these important features influence the failure rotation.

Besides, Figure 15 also indicates that different input features are decisive in the learning phase for predicting the initial failure rotation, comparing to the multi-output RF model. Therefore, the reason why the multi-output RF model cannot achieve good predictive performance on the initial failure rotation is that different input features are decisive in determining this output, while the developed multi-output RF model relies on other features and neglect the effects of input features shown in Figure 15 on the initial failure rotation.

5. Discussion

As demonstrated in section 2.2.1, prior studies have noted the importance and potential of using the metal-modeling techniques in solving engineering problems in the AEC industry during the past few decades, yet very little was found in the literature on investigating how can meta-modeling techniques can be adapted to the field of the wind turbine foundation design. On this premise, the main contribution of conducting the presented study is providing an opportunity to determine how and to which extent can the meta-model be used to streamline the wind turbine design process to obtain the optimal design solution, by providing surrogates to replace the FEA in the traditional static method.

Based on results represented in section 4.3, this study found that the developed multi-output RF model was recognized as the most favorable choice, considering the computational speed and accuracy. This finding differs from that of Kayri et al. (2017) who found that the neural network can be regarded as the best option for solving big and complex data mining problems by comparing RF and the neural network, but it is broadly consistent with the earlier study conducted by Oliveira et al. (2012), in which RF was examined to have better performance than the neural network.

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 anchors contributes the least.

Surprisingly, using single-output machine learning models can generally provide an improved prediction on every single output. Therefore, by developing 8 different single-output models to independently predict the moment-rotation behavior of the wind turbine foundation. This finding is contrary to that of Borchani et al. (2015) 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 to this finding might be that the currently used dataset is not sufficiently large to provide enough inferences regarding correlations between selected features and these 8 outputs. Comparing to the single-output machine learning models, which are developed to predict only one single target, 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 machine learning models.

Furthermore, although each single-output model can provide better accuracy, thus having the potential to conduct the problem transformation methods to solve this multi-output regression problem, it is still recommended to apply the multi-output machine learning models to predict all the outputs simultaneously, given the fact that developing 8 different single-output models is more time-consuming compared to generating one multi-output development model, as shown in Table 8.

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, by developing the machine-learning-based meta-models, it 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 the 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 aim of the study is to investigate how and to which extent can the machine-learning based-model be developed and utilized as the substitution of the FEA, which hinders the design optimization process of the wind turbine foundations. In order to do that, a meta-model developing process was proposed, mainly including the dataset establishment and the machine learning models development.

In this study, two multi-output machine learning algorithms were selected, namely the multi-output RF and multi-output FFNN, and optimized by using the proposed GA method 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 named ABT. In general, the multi-output RF model shows better performance compared to other developed models, regarding the accuracy and computational time.

However, because currently only a limited number of design variables were considered in this research as input features, further study is required to evaluate this method in a broader dataset. This would require a big 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 and 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 are 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.

References

Abo-Hamad, W., & Arisha, A. (2011). Simulation Optimisation Methods in Supply Chain Applications A Review. EBSCOhost. *Irish Journal of Management, 30*(2), 95-124.

Ali, J., Khan, R., Ahmad, N., & Maqsood, I. (2012). Random forests and decision trees. *nternational Journal of Computer Science Issues (IJCSI), 9*(5), 272.

Alpaydin, E. (2010). *Introduction to machine learning (2nd ed., Ser. Adaptive computation and machine learning).* MIT Press.

An, N., Zhao, W., Wang, J., Shang, D., & Zhao, E. (2013). Using multi-output feedforward neural network with empirical mode decomposition based signal filtering for electricity demand forecasting. *Energy, 49*, 279-288.

Arabasadi, Z., Alizadehsani, R., Roshanzamir, M., Moosaei, H., & Yarifard, A. A. (2017). Computer aided decision making for heart disease detection using hybrid neural network-Genetic algorithm. *Computer methods and programs in biomedicine, 141*, 19-26.

Arora, J. (2012). *Introduction to Optimum Design (3rd edition).* Elsevier.

Bakalis, K., Fragiadakis, M., & Vamvatsikos, D. (2017). Surrogate modeling for the seismic performance assessment of liquid storage tanks. *Journal of Structural Engineering, 143*(4), 04016199.

Bilgili, M., Yasar, A., & Simsek, E. (2011). Offshore wind power development in Europe and its comparison with onshore counterpart. *Renewable and Sustainable Energy Reviews, 15(2)*, 905-915.

Borchani, H., Varando, G., Bielza, C., & Larrañaga, P. (2015). A survey on multi-output. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 5*(5), 216-233.

Bouktif, S., Fiaz, A., Ouni, A., & Serhani, M. A. (2018). Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches. *Energies, 11*(7), 1636.

Breiman, L. (2001). Random Forests. *Machine Learning, 45*(1), 5-32.

Breiman, L. (2002). *Manual on setting up, using, and understanding random forests v3. 1.* CA, USA, 1, 58: Statistics Department University of California Berkeley.

Burton, T., Jenkins, N., Sharpe, D., & Bossanyi, E. (2011). *Wind Energy Handbook.* Wiley.

Chan, W., Chow, Y., & Liu, L. (1995). Neural-network—an alternative to pile driving formulas. *Comput Geotech, 17*(2), 135–156.

Chen, J., Gao, X., Hu, Y., Zeng, Z., & Liu, Y. (2019). A meta-model-based optimization approach for fast and reliable calibration of building energy models. *Energy, 188*, 116046.

Cherkassky, V., & Mulier, F. M. (2007). *Learning from data: concepts, theory, and methods.* John Wiley & Sons.

Dai, B., Gu, C., Zhao, E., & Qin, X. (2018). Statistical model optimized random forest regression model for concrete dam deformation monitoring. *Structural Control and Health Monitoring, 25*(6), e2170.

Dasari, S. K., Cheddad, A., & Andersson, P. (2019). Random Forest Surrogate Models to Support Design Space Exploration in Aerospace Use-Case. *IFIP International Conference on Artificial Intelligence Applications and Innovations* (pp. 532-544). Cham: Springer.

Dietterich, T. G. (2000). Ensemble methods in machine learning. *International workshop on multiple classifier systems* (pp. 1-15). Berlin, Heidelberg: Springer.

Du, D., Li, K., & Fei, M. (2010). A fast multi-output RBF neural network construction method. *Neurocomputing, 73*(10-12), 2196-2202.

Engelbrecht, A. P. (2006). *Fundamentals of Computational Swarm Intelligence.* NJ: John Wiley & Sons.

Gao, X., & Lee, G. M. (2019). Moment-based rental prediction for bicycle-sharing transportation systems using a hybrid genetic algorithm and machine learning. *Computers & Industrial Engineering, 128*, 60-69.

Ghosh , J., Padgett, J. E., & Dueñas-Osorio, L. (2013). Surrogate modeling and failure surface visualization for efficient seismic vulnerability assessment of highway bridges. *Probabilistic Engineering Mechanics, 34*, 189-199.

Gosavi, A. (2010). Simulation-based Optimization – Parametric Optimization Techniques and Reinforcement Learning. *Kluwer Academic*.

Günaydin , H. M., & Doğan , S. Z. (2004). A neural network approach for early cost estimation of structural systems of buildings. *International Journal of Project Management, 22*(7), 595-602.

Guo, J., Wang, Q., Li, Y., & Liu, P. (2020). Façade defects classification from imbalanced dataset using meta learning-based convolutional neural network. *Computer-Aided Civil and Infrastructure Engineering*.

Hayati, M., & Shirvany, Y. (2007). Artificial neural network approach for short term load forecasting for Illam region. *World Academy of Science, Engineering and Technology, 28*, 280-284.

Ismail , A., & Jeng , D. (2011). Modelling load-settlement behaviour of piles using high-order neural network (HON-PILE model). *Eng Appl Artif Intell, 24*(5), 813-821.

Jin, Y. (2011). Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation, 1*(2), 61-70.

Jung, Y. (2018). Multiple predicting K-fold cross-validation for model selection. *Journal of Nonparametric Statistics, 30*(1), 197-215.

Kayri, M., Kayri , I., & Gencoglu, M. T. (2017). The performance comparison of Multiple Linear Regression, Random Forest and Artificial Neural Network by using photovoltaic and atmospheric data. *2017 14th International Conference on Engineering of Modern Electric Systems (EMES)* (pp. 1-4). IEEE.

Kocev, D., Džeroski, S., White, M. D., Newell, G. R., & Griffioen, P. (2009). Using single-and multi-target regression trees and ensembles to model a compound index of vegetation condition. *Ecological Modelling, 220*(8), 1159-1168.

Kocev, D., Vens, C., Struyf, J., & Džeroski, S. (2007). Ensembles of multi-objective decision trees. *European conference on machine learning* (pp. 624-631). Berlin, Heidelberg: Springer.

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering, 160*, 3-24.

Kumar, K. V., Ravi, V., Carr, M., & Kiran, N. R. (2008). Software development cost estimation using wavelet neural networks. *Journal of Systems and Software, 81*(11), 1853-1867.

Li , A., Khoo , S., Lyamin , A., & Wang , Y. (2016). Rock slope stability analyses using extreme learning neural network and terminal steepest descent algorithm. *Autom Constr, 65*, 42-50.

Loubser, P. B., & Jacobs, A. R. (2016). Optimised design of wind turbine gravity foundations. *Insights and Innovations in Structural Engineering, Mechanics and Computation: Proceedings of the 6th International Conference on Structural Engineering,* (pp. 5-7). Mechanics and Computation.

Mardfekri, M., & Gardoni, P. (2013). Probabilistic demand models and fragility estimates for offshore wind turbine support structures. *Eng Struct, 52*, 478-487.

Mey, F., & Diesendorf, M. (2018). Who owns an energy transition? Strategic action fields and community wind energy in Denmark. *Energy Research & Social Science, 35*, 108-117.

Moayedi, H., Gör, M., Khari, M., Foong, L. K., Bahiraei, M., & Bui, D. T. (2020). Hybridizing four wise neural-metaheuristic paradigms in predicting soil shear strength. *Measurement: Journal of the International Measurement Confederation, 156*, 107576.

Moayedi, H., Mosallanezhad, M., & Nazir, R. (2017). 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*(3), 150–154.

Muskulus, M., & Schafhirt, S. (2014). Design optimization of wind turbine support Structures-A Review. *Journal of Ocean and Wind Energy, 1*(1), 12-22.

Naik, G. M., & Kumar, M. (2013). Project Cost and duration optimization Using Soft Computing Techniques. *Journal of Advanced Management Science, 1*(3), 299-303.

Nicholson, J. C. (2011). *Design of wind turbine tower and foundation systems: optimization approach.* University of Iowa.

Oliveira, A. L., Barga, P. L., Lima, R. M., & Cornélio, M. L. (2010). GA based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation. *information and Software Technology, 52*(11), 1155-1166.

Olivera, S., Oehler, F., San-Miguel-Ayanz, J., Camia, A., & Pereira, J. M. (2012). Modeling Spatial Patterns of Fire Occurrence in Mediterranean Europe using Multiple Regression and Random Forest. *Forest Ecology and Management, 275*, 117-129.

Oswell, J. M., Mitchell, M., Chalmers, G., & Mackinven, H. (2010). Design, construction and initial performance of wind turbine foundations in Antarctica. *Proceedings GEO*, 997-1003.

Querin, O. M., Victoria, M., Gordoa, C. A., Ansola, R., & Martí, P. (2017). *Topology design methods for structural optimization.* Butterworth-Heinemann.

Quirante, N., Javaloyes, J., & Caballero, J. A. (2015). Rigorous design of distillation columns using surrogate models based on K riging interpolation. *AIChE Journal, 61*(7), 2169-2187.

Razavi, S., Tolson, B. A., & Burn, D. H. (2012). Review of surrogate modeling in water resources. *Water Resources Research, 48*(7), W07401.

Rouss, V., & Charon, W. (2008). Multi-input and multi-output neural model of the mechanical nonlinear behaviour of a PEM fuel cell system. *Journal of Power Sources, 175*(1), 1-17.

Roy, M. H., & Larocque, D. (2012). Robustness of random forests for regression. *Journal of Nonparametric Statistics, 24*(4), 993-1006.

Roy, R. (2010). *A Primer on the Taguchi Method.* Soc of Mfg Engrs.

Samui , P., & Sitharam , T. (2010). Site characterization model using least-square support vector machine and relevance vector machine based on corrected SPT data (N-c). *Int J Numer Anal Methods Geomech, 34*(7), 755–770.

Sherif, S. A., Barbir, F., & Veziroglu, T. N. (2005). Wind energy and the hydrogen economy—review of the technology. . *Solar Energy, 78(5)*, 647–660.

Spall, J. (2003). *Introduction to Stochastic Search and Optimization – Estimation, Simulation, and Control.* Wiley.

Spyromitros-Xioufis, E., Tsoumakas, G., Groves, W., & Vlahavas, I. (2012). Multi-label classification methods for multi-target regression. *arXiv preprint arXiv:1211.6581*, 1159-1168.

Stegen, K. S., & Seel, M. (2013). The winds of change: How wind firms assess Germany's energy transition. *Energy Policy, 61*, 1481-1489.

Struyf, J., & Džeroski, S. (2006). Constraint based induction of multi-objective regression trees. *International Workshop on Knowledge Discovery in Inductive Databases* (pp. 222-233). Berlin, Heidelberg: Springer.

Thiry, A., Bair, F., Buldgen, L., Raboni, G., & Rigo, P. (2011). Optimization of Monopile Offshore Wind Structures. *Adv Mar Struct, Taylor & Francis*, 633-642.

Urbanek, T., Prokopova, Z., Silhavy, R., & Vesela, V. (2015). Prediction accuracy measurements as a fitness function for software effort estimation. *SpringerPlus, 4*(1), 778.

Van Der Tempel, J. (2006). *Design of support structures for offshore wind turbines.* PhD Thesis, TU Delft, the Netherlands.

Vorpahl, F., Schwarze, H., Fischer, T., Seidel, M., & Jonkman, J. (2013). Offshore Wind Turbine Environment, Loads, Simulation, and Design. *WIREs Energy Environ, 2*, 548-570.

WindEurope. (2020). *Wind energy in Europe in 2019 - Trends and statistics.* Brussels, Belgium: WindEurope. Retrieved from https://windeurope.org/data-and-analysis/product/?id=59

Wong, T. T. (2015). Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition, 48*(9), 2839-2846.

Worden, K., & Manson, G. (2007). The application of machine learning to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 365*(1851), 515-537.

Wu , A., Zeng , Z., & Fu, C. (2014). Data mining paradigm based on functional networks with applications in landslide prediction. *Proceedings of the 2014 international joint conference on neural networks* (pp. 2826–2830). New York: IEEE.

Zheng, H., Moosavi, V., & Akbarzadeh, M. (2020). Machine learning assisted evaluations in structural design and construction. *Automation in Construction, 119*, 103346.