Automation in Construction 177 (2025) 106287

Contents lists available at ScienceDirect

Automation in Construction

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

Generative AI-augmented offshore jacket design: Integrated approach for mixed tabular data generation under scarcity and imbalance

Emmanouil Panagiotou [a,b,*], Han Qian [c], Steffen Marx [c], Eirini Ntoutsi [b]

[a] Institute of Computer Science, Freie Universität Berlin, Kaiserswerther Str. 16-18, Berlin, 14195, Brandenburg, Germany
[b] Research Institute CODE, Bundeswehr Universität Munich, Werner-Heisenberg-Weg 39, Munich, 85577, Neubiberg, Germany
[c] Institute of Concrete Structures, Technische Universität Dresden, Helmholtzstr. 10, Dresden, 01062, Saxony, Germany

ARTICLE INFO

Keywords:
Artificial intelligence
Generative AI
Machine learning
Multi-objective optimization
Mixed tabular data
Data scarcity
Data imbalance
Offshore wind turbines
Industrial design
Offshore jacket substructure

ABSTRACT

Generative Artificial Intelligence (AI) has found various applications in domains like computer vision and natural language processing. However, limited research exists in the engineering domain, where prevailing challenges involve mixed tabular data, data scarcity, and imbalances. This paper focuses on generating synthetic offshore jacket designs to improve the data quality of a scarce and imbalanced existing dataset. Data quality is quantified by evaluating the machine-learning efficiency of the synthetic data on a domain-specific downstream task.

An integrated method is proposed for generating jacket designs, combining modern data-driven techniques with traditional multi-objective-driven approaches. The method addresses challenges related to mixed attributes, data scarcity, and class imbalances. Experimental results demonstrate improved predictive performance on the downstream task when models are trained on synthetic data compared to using only real data. These findings contribute to the advancement of generative AI in offshore engineering and related fields, offering valuable insights and potential applications.

1. Introduction

The evolution of Artificial Intelligence (AI) has seen a transformative shift from conventional approaches to data-driven methodologies. Initially, rule-based searching algorithms were used to solve specific tasks, while modern data-driven AI approaches can leverage large datasets to extract patterns and insights across diverse domains such as medicine, finance, arts, and engineering. Recently, generative methods have demonstrated exceptional capabilities in tasks previously addressed solely by humans, like image and text generation, while also proving effective in improving data quality for AI in various downstream tasks [1]. However, these advancements often rely on the use of very large models with billions of parameters trained on extensive datasets [2,3].

In certain domains, like engineering and construction, there is a shortage of available data (*data scarcity*), and moreover, *population imbalances* are common. For example, in the sector of offshore wind farm construction, design specifications are often withheld by companies, exacerbating the issue of data scarcity. Furthermore, many real-world wind turbine substructures adhere to specific design types, such as monopiles or four-legged X-brace jackets [4], resulting in an imbalanced (biased) population towards certain feature values or combinations thereof. These data characteristics, i.e., scarcity and imbalance, frequently coincide, posing challenges to the application of advanced generative AI techniques and have hindered research in this direction [5]. Additionally, in most real-life applications, and especially in engineering, the most common data modality is tabular data of mixed nature, containing both discrete and continuous features [6]. This particularity poses a problem for many deep generative methods that often assume a common continuous data type. Some studies explore the use of generative AI in structural engineering, particularly in structural health monitoring (SHM) [7]. In this domain, despite the vast amounts of data that can be gathered from sources like sensors, labeled data are limited. However, most approaches focus on oversampling via generative adversarial networks (GANs) [8,9].

Our work is directed towards learning to generate synthetic mixed tabular data under data scarcity and imbalances. Our goal is to augment the existing data, improving the predictive performance and generalization of predictive models on a domain-specific downstream task. We focus on the offshore engineering domain using a small dataset of real offshore jacket substructure designs. An offshore jacket substructure is a type of foundation commonly used for offshore wind turbines,

[*] Corresponding author at: Institute of Computer Science, Freie Universität Berlin, Kaiserswerther Str. 16-18, Berlin, 14195, Brandenburg, Germany.
E-mail address: emmanouil.panagiotou@fu-berlin.de (E. Panagiotou).

https://doi.org/10.1016/j.autcon.2025.106287
Received 3 January 2024; Received in revised form 14 May 2025; Accepted 14 May 2025
Available online 2 June 2025
0926-5805/© 2025 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

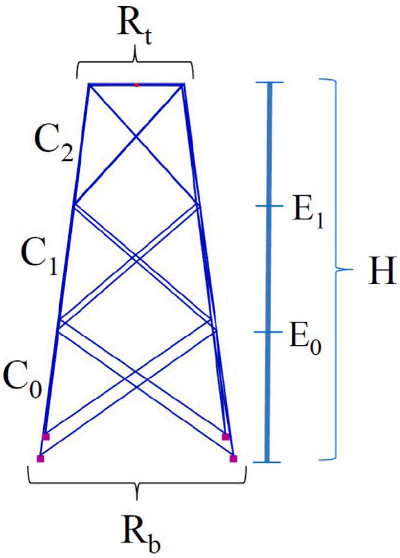


Fig. 1. Four-legged jacket structure with four layers, along with its various features.

especially in deeper waters. The term “jacket” refers to the lattice-like steel framework that makes up the substructure (see Fig. 1). This framework extends from the seabed to the surface and is anchored into the seabed. The jacket’s lattice structure is designed to withstand the marine environment, including waves, currents, and wind loads, as well as the weight and dynamic loads of the wind turbine itself. This domain is fitting to our problem setting, given that the dataset contains only 100 jacket designs that exhibit extreme imbalances for specific features (e.g. braces) [10].

In our study, we examine purely data-driven generative methods with major differences in the modeling strategy. Specifically, a neural network approach, i.e., Tabular Variational Autoencoder (TVAE [11]), a probabilistic approach, i.e,. Mixed Deep Gaussian Mixture Model (MDGMM [12]), and a transformer-based Large Language Model (LLM [1]) approach. In [11], CTGAN, a GAN-based generative model for tabular data, is introduced alongside TVAE. However, since TVAE has been shown to achieve better performance in tabular data generation [11], we use it in our experiments instead of CTGAN. Nonetheless, data-driven methods have limitations coming from the data, as they can only generate in the confines of the existing domain, thereby perpetuating existing imbalances. Therefore, while data-driven methods can address data scarcity, they may not effectively mitigate imbalances. To search for more diverse solutions, we use a conventional multi-objective Genetic Algorithm (GA) [13] that inherently generates multiple trade-off solutions. Our objectives are design-driving domain-specific fitness functions such as the total cost and load-bearing capacity of a structure. Since the search space of GAs is not constrained by the real data, we extend it by expanding the domain of certain imbalanced features, i.e., brace types, using domain knowledge. However, to control the plausibility of the GA-generated designs with respect to the real data, we propose a plausibility objective and threshold, assisting in pushing solutions towards the real data manifold. By combining the data-driven and objective-driven approaches, we design our integrated generative method that covers a wider region of the existing space, thereby improving the quality and diversity of the synthetic data.

We design our experiments following similar works, comparing all methods in terms of Machine Learning Efficiency (MLE) on a domain-specific downstream task, i.e., predicting the fitness of structures based on their input features. It is important to note that we do not generate synthetic targets for the downstream task. Instead, targets such as the load-bearing capacity of a structure are acquired via simulation or analytical computation. Consequently, we obtain real ground truth values for both real and synthetic structures. To quantify the synthetic data quality, we measure the improvement in MLE of various predictive models when trained on real data vs synthetic data. Results show that the more lightweight MDGMM outperforms other data-driven methods at generating synthetic instances close to the original data. Additionally, we find that our proposed plausibility objective is a successful controlling parameter between the plausibility and diversity of the GA-generated designs. Ultimately, our integrated generative method that combines both data-driven and objective-driven approaches fulfills our goal of improved MLE, while also aiding towards generalization beyond the real dataset limitations.

Despite significant progress in AI-based generative modeling, there remains a gap in applying these methods to low-resource, imbalanced, and mixed-tabular domains like structural engineering. This research addresses that gap by offering a comprehensive solution that combines data-driven and objective-driven approaches to synthesize data and enhance the performance of predictive models. The findings and insights presented in this paper have the potential to significantly advance the application of generative AI in offshore engineering and related fields. The key contributions of this paper are as follows:

- We conduct a comparative study of three generative modeling paradigms, i.e., neural-network-based, probabilistic, and LLM-based, on mixed tabular data under data scarcity and imbalance.
- We propose an integrated generative framework that combines data-driven generation with a domain-specific, objective-driven genetic algorithm, enhanced by a plausibility control mechanism.
- We apply our method to a real-world offshore jacket substructure dataset and demonstrate improvements in synthetic data quality and downstream model generalization.

The rest of this paper is structured as follows: Section 2 outlines our problem formulation and provides an overview of our dataset. In Section 3, we comprehensively discuss all data-driven and objective-driven methods employed in our research. Section 4 delves into the details of our integrated methodology, while Section 5 presents the results of our experiments. Furthermore, in Section 6, we explore potential applications of our approach in real-world scenarios. Finally, Section 7 concludes the paper with limitations and suggestions for future work. The codebase and the dataset are accessible through the following GitHub repository link, enabling the reproduction of all results and figures presented in this paper. github.com/Panagiotou/Offshore_Jacket_Design_Augmentation.git

2. Basic notions and problem formulation

Our goal is to generate realistic synthetic designs, expanding beyond real dataset limitations to improve the generalization of ML models in downstream tasks, i.e., predicting the load-bearing capacity and the total cost of a structure. Our analysis is based on the dataset of real jacket structures first introduced in [10]. The dataset, consisting of a mere 100 instances, poses challenges for data-driven generative AI due to its limited size and bias towards certain feature values (Section 2.1). To expand beyond the real data limitations, we extend the design space as presented in Section 2.2.

2.1. Real dataset of existing designs

For the real dataset [10], data collection is initiated from the 4C Offshore database [4], yet a significant portion of the missing structural details had to be sourced from, e.g., reports of offshore wind farm projects, images, and news reports. The dataset comprises 100 jacket designs sourced from more than 90 wind farms, primarily located in Asia, Europe, and the United States. It covers jacket substructures from various site-specific factors such as water depth, mean wind speed, wave height, etc.

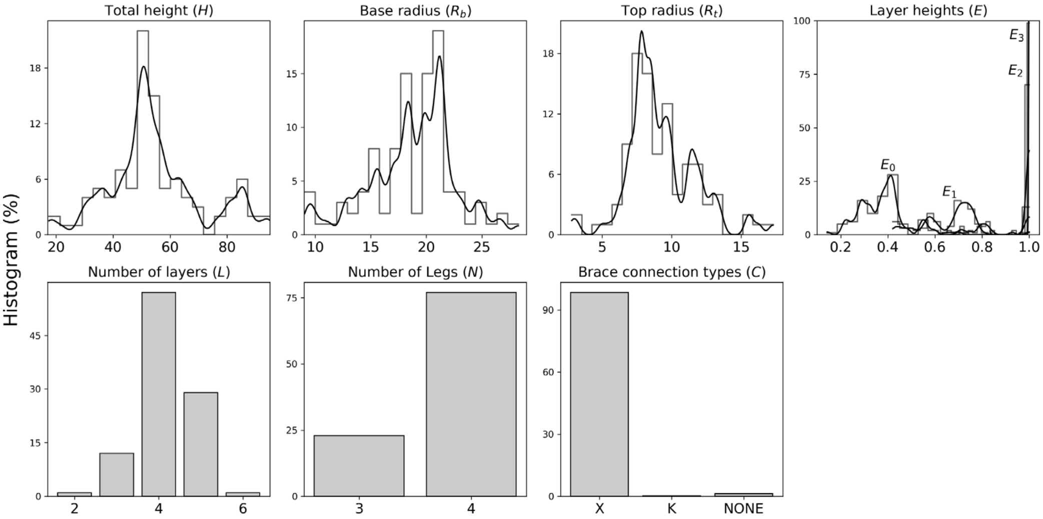


Fig. 2. Dataset distributions for all features. The y-axis represents the percentage of data.

Table 1
Feature representation of jacket structures.

Feature	Description	Type	Value domain (real data)	Value domain (extended)
H	Total height	Continuous	(17 – 95)	–
R_b	Base radius	Continuous	(9 – 29)	–
R_t	Top radius	Continuous	(2 – 18)	–
E	Layer heights (% of total height): $\{E_0, E_1, \dots\}$	Continuous (list)	(0 – 1)	–
L	Number of layers	Discrete ordinal	$\{2 - 6\}$	–
N	Number of legs	Discrete ordinal	$\{3, 4\}$	–
C	Brace connection types (between layers): $\{C_0, C_1, \dots\}$	Discrete nominal (list)	$\{X, K, NONE\}$	++ $\{H, Z, IZ, ZH, IZH, XH\}$

A tabular representation of the dataset is provided in Table 1. Following [14], each jacket design is represented in terms of seven dimensions covering the essential structural information, e.g., total height, layer heights, top and bottom widths, bracing types, leg number, etc. Fig. 1 illustrates an instance of such a structure along with its various features. It is evident that the feature space is mixed, consisting of both continuous and discrete features. The latter can be further categorized into ordinal (with a defined order, like the number of layers L) or nominal (no natural ordering is assumed, like connection types C). In the same table, we also provide the value domain of the different attributes. We distinguish between the value domain based on the dataset of real structures and the extended value domain as defined by domain experts and literature. As already mentioned, our goal is to learn to generate designs beyond the “boundaries” of the available dataset, and for this, we will rely on the expert/domain boundaries.

The layer heights (E) and brace connections (C) are lists of values that can change in size, contingent on the number of layers (L) of the structure. For example, the structure of Fig. 1 with four layers has three braces and a list of two layer heights.

Additionally, the distribution of the dataset in terms of the different characteristics/features is presented in Fig. 2. Examining the distributions reveals that we are not only dealing with *data scarcity* but also imbalance in certain features. A characteristic example of extreme *imbalance* is the brace types (C), with almost all real structures in our dataset (99%) containing solely X braces [4]. We extend this feature as presented in Section 2.2 to increase the coverage of potential designs.

To summarize, the real dataset poses various challenges for AI, including data scarcity (only 100 instances), imbalanced representations favoring certain feature values, for example, the X brace connection type, and a mixed attribute space consisting of four continuous and three categorical features.

2.2. Extended design space

To overcome the limitations of the real data, we extend the design space to explore a wider range of structures. More concretely, we observe a significant bias in the value domain of the brace type attribute (C), predominantly leaning towards X braces.

The prevalence of four-legged X-brace jackets suggests that this design is effective, but it does not necessarily imply that it is the sole or optimal choice, particularly when considering the diverse criteria for structural integrity. Looking ahead, as the demand for larger structures capable of bearing heavier loads at reduced costs grows, engineers in academia and industry are increasingly dedicated to exploring alternative designs [15], emphasizing a wider range of potential braces such as K, Z, etc [16,17]. However, such alternative designs are not yet part of our dataset of real structures, as the transition from theoretical exploration to practical implementation requires time and rigorous testing. Nonetheless, we extend the design space for the brace-type feature (C) considering its extreme bias towards X-braces, to explore the numerous potential brace types (Fig. 3).

No such extensions are made for other features, given their adequate coverage and less pronounced imbalances. Nonetheless, similar extensions can be seamlessly implemented in future works, such as extending the total height and number of layers for generating larger structures.

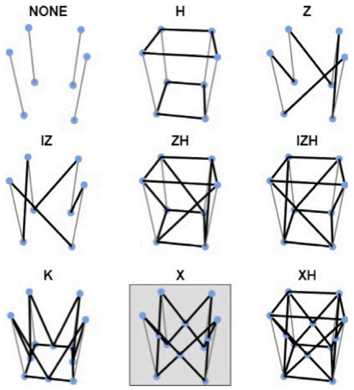


Fig. 3. Potential connecting brace types (C) between layers. The X brace connection type (highlighted) is overwhelmingly dominant in the dataset.

Table 2
Objective space: Design-driving performance indicators.

Objective	Definition	Type
Cost	$\sum \|edge_i\|$	Analytically computed
Compression (S_C)	Top load (6835 kN)	Simulated
Pushover (S_P)	Side load (16.64 kN)	Simulated
Tortion (S_T)	Torque load (100 kN m)	Simulated
Wave (S_W)	$F_{Morisson}$ load	Simulated
Combined (S_{comb})	Combined load	Simulated

2.3. Objective space

The goodness or quality of a structure depends on various objectives, which are design-driving performance/fitness indicators of a structure. An ideal structure should depict low cost while having a large load-bearing capacity, for various load cases. Many more objectives can be important, stemming from the entire life cycle of a structure.

An overview of the objectives and their definitions is shown in Table 2, explanations follow hereafter.

Cost objective: One of the major objectives in any engineering project is the estimation of the total *Cost*. We estimate the cost of a structure analytically by assuming that it is proportional to the overall material usage. Hence, we derive an estimate by summing the total length of all edges.

Load-bearing capacity objectives: The load-bearing capacity is the primary function of a substructure, and therefore, the corresponding objectives are of great importance. We estimate the load-bearing capacity via finite element modeling (FEM) simulation (see Section 2.4) that stress-tests the structures under various load cases of reference wind energy projects [18]. More specifically, the *Compression* (S_C), *Pushover* (S_P), and *Tortion* (S_T) tests represent external loads from the RNA (Rotor-Nacelle Assembly) and the tower of the wind turbine. The *Wave* (S_W) test simulates a wave impact on the structure, while the *Combined* (S_{comb}) test applies all previous loads simultaneously. A graphical representation of the resulting simulations is depicted in Fig. 4.

The exact external loads of Table 2 for the *compression*, *pushover*, and *tortion* are assumed to be constant for all structures [15,19] and extracted from reference wind energy projects [18]. The wave load case is more complicated since it can depend on site-specific information. We follow [14] and compute the load dynamically for each structure, applying it on both legs of one side of the structure at a certain height of impact $H_w = 0.57 \cdot H$. This relative height (H_w) is estimated as the average water-depth to structure-height ratio. The load value is calculated using the Morison equation [20], assuming a constant wave of height 1.19 m and period 6.59 s. All wave characteristics are averaged from our datasets' cite-specific conditions [10]. We assess the effect of all load cases by measuring the *maximum von Mises stress* [21] on the bottom nodes of the structures.

2.4. Numerical simulation (FEM)

We perform all structural tests using finite element simulations, a standard practice for static structural analysis of jacket designs [15,22]. The finite element method (FEM) solves differential equations arising in engineering problems, such as structural analysis by breaking down a larger system (like the entire structure) into smaller finite elements. These elements enable simpler local approximations of the more complex global differential equations. The numerical results are interpreted to provide useful information about the original problem, such as the structure's distribution of stresses and strains based on specific boundary conditions and loading.

For consistent structural testing across real and synthetic structures under various load cases, we rely on the open-source finite element simulation framework *OpenSeesPy* [23]. This framework, widely used for stress-testing structures in similar applications [24], ensures robust and efficient structural evaluation. The complete framework for the structural evaluation of any given jacket structure is provided in our codebase (a link to the GitHub repository is provided in the introduction).

3. Related work

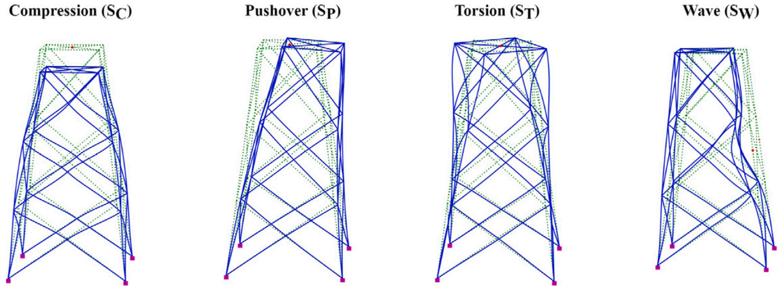
This section provides coverage of the related work. The existing literature can be categorized into two main approaches: data-driven and objective-driven. Data-driven methods focus on learning the underlying data distribution, while objective-driven methods aim at optimizing solutions based on specific objectives. We analyze both categories below, emphasizing the most relevant methods for our work.

3.1. Data-driven methods

Data-driven methods leverage existing datasets to train models that can generate new, realistic data. A large number of methods, from autoencoders to adversarial and transformers-based models, have been proposed in the last few years. Yet, generative models tailored for handling tabular data are constrained due to the unique characteristics of such data, as mentioned earlier. Here we present three such methods that are most relevant to our work, namely, a neural network-driven autoencoder, a probabilistic Gaussian mixture model, and an approach based on a pre-trained large language model.

TVAE: Variational Autoencoder (VAE) models [25], like conventional autoencoders, capture a mapping from the input data to a (typically lower dimensional) latent space, and inversely, reconstruct data samples from that space. The latent space in VAEs is regularized to a target probabilistic distribution (e.g., Gaussian) for smooth generative sampling. The learning objective is to maximize the likelihood of generating accurate reconstructions. This is achieved by simultaneously minimizing the reconstruction loss of the input and the reconstructed sample, and the regularization term (namely, the Kullback–Leibler divergence) between the latent distribution and the target distribution. After training, synthetic data is generated through the decoder by sampling random samples from the latent distribution.

Relevant to our work is the recently proposed tabular VAE (TVAE) [11] which overcomes the limitations of existing deep generative methods that are unable to properly model mixtures of both discrete and continuous data. Fully connected networks are used for the encoder–decoder networks to capture possible feature correlations. Continuous variables are assumed to follow a mixture of Gaussian distributions identified by a variational Gaussian mixture model (VGM) [26], while discrete variables follow a categorical probability distribution (PMF). The decoder model represents the conditional distribution $p(x|z)$ of the input x given the lower-dimensional latent representation z. Since z follows a normal distribution $z \sim \mathcal{N}(\mu, \sigma)$ of mean μ and standard deviation σ, it can be used to sample in the latent space and reconstruct novel data points in the input space through the decoder.

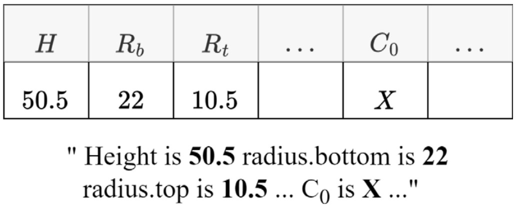

Compression (S_C)

Pushover (S_P)

Torsion (S_T)

Wave (S_W)

Fig. 4. Visualization of the effect of structural load cases on jacket structures. Deformations are enhanced for visualization purposes.



H	R_b	R_t	$\ldots$	C_0	$\ldots$
50.5	22	10.5		X	

" Height is **50.5** radius.bottom is **22** radius.top is **10.5** ... C_0 is **X** ..."

Fig. 5. Example textual encoding of an instance from our dataset.

MDGMM: Gaussian mixture models (GMM), originally a probabilistic clustering approach, model the input data as a mixture of Gaussians by finding the optimal distribution parameters via the Expectation–Maximization (EM) algorithm. Synthetic data can be generated by sampling from the fitted mixture model. Nonetheless, mixed data inherently do not follow Gaussian distributions, and therefore, GMMs are unsuitable in this context. To address this limitation, a Mixed Deep GMM (MDGMM) was introduced in [12]. Deep GMMs, similar to neural networks, have a multi-layer architecture of simple functions that collectively capture complex patterns in the data. Each layer consists of a mixture of factor analyzers (MFA) [27] that progressively reduce the dimensions of the input until a lower-dimensional latent representation is reached. Under the MDGMM discrete variables are unified into a continuous latent space via a Generalized Linear Latent Variable Model (GLLVM) [28] by assuming exponential link functions between the discrete input and the continuous latent variables (mixture of Gaussians). This gives more information to the model about the statistical nature of the discrete variables, which can be crucial when dealing with data scarcity. For example, a Bernoulli distribution can be used for binary data or a binomial distribution for ordinal data [29].

The MDGMM approach is suitable for synthetic data generation [30] as it is fully invertible by design under the *conditional independence assumption* (GLLVM) for which variables are mutually independent given the latent variables. Specifically, the Bayes rule can be used on the link functions f with learnable parameters θ, giving:

$$f(x|\theta) = \frac{f(z|\theta)f(x|z,\theta)}{f(z|x,\theta)} \propto f(z|\theta)\prod_{j\in|x|} f(x_j|z,\theta) \tag{1}$$

where x represents the input observations, z the samples drawn from the latent space, and $|x|$ denotes the set of feature indices in the observation vector x.

By sampling z in the latent space through the MDGMM, synthetic samples are generated from $f(x|z,\theta)$.

LLM: Recently, Large Language Models (LLM) have been used to generate textual data with great success. This is due to their ability to recurrently predict the optimal next token based on a conditional input sequence of previous language tokens. Consequently, pre-trained LLMs have been proposed for tabular data generation [1] after appropriately transforming the tabular instances into syntactical sentences (textual encoding). An example textual encoding from our dataset (Section 2.1) is presented in Fig. 5.

To adapt a large pre-trained LLM to the task-specific tabular data generation method, fine-tuning is performed [1,31,32] on the encoded dataset. The fine-tuned LLM q, gives a categorical output distribution $z = q(w_1,\ldots,w_{k-1})$ over potential follow-up tokens. Generation of synthetic data is achieved as in [1] employing the following sampling strategy from the output z using a temperature parameter T:

$$P(\omega|w_1,\ldots,w_{k-1}) = \frac{e^{(z_\omega/T)}}{\sum_{\omega'\in W} e^{(z_{\omega'}/T)}} \tag{2}$$

here, z_ω is the unnormalized logit score (i.e., raw model output) for token ω, W is the full vocabulary of candidate tokens. The temperature parameter $T > 0$ controls the randomness of the sampling process: lower values of T (< 1) make the distribution sharper (more deterministic), favoring high-probability tokens, while higher values of T (> 1) flatten the distribution, encouraging more diverse outputs. We us the default value of $T = 0.7$, as it has been empirically shown to perform well in [1]. Then, the probability P of the next token ω is estimated by conditional weighted sampling based on an input sequence $w_1,\ldots,w_{k-1}$. This can be performed with any arbitrary conditioning, meaning that the sampling procedure can start from any feature, e.g. for our data *"Layers are"* → 4 or "*Brace type is*" → X, etc.

3.2. Objective-driven methods

Although solely data-driven methods have found great success in generating synthetic data, as previously discussed, they are unable to synthesize outside of the domain dictated by the input data. On the other hand, objective-driven methods, such as Genetic Algorithms (GA), solve optimization problems in a search-based manner, progressively evolving a population of candidates. Because GAs are guided solely by fitness objectives, they can cover a space larger than the existing feature domain of the data. Therefore, defining specific objectives for a given domain allows the exploration of solutions with diverse characteristics. The resulting solutions are synthetic instances that can differ from the existing data manifold while still being optimal with regard to the domain-specific objectives.

Most GAs are single-objective, meaning that feedback on the goodness of candidate solutions is provided through a single fitness function. The goal in this setting is to converge towards a single globally optimal solution. Similar optimization methods have been used in the wind energy domain to optimize specific parameters of offshore substructures [22,33,34]. Nonetheless, the goal of this work is to generate many varying solutions to mitigate scarcity and imbalances in the existing dataset. Therefore, following [14], we formulate our augmentation approach as a multi-objective optimization (MOO) problem and solve it using a multi-objective (MO) genetic algorithm. This results in a Pareto-optimal set of multiple trade-off solutions (synthetic data). Several multi-objective GAs and their various iterations have been proposed, differing in their selection strategy, fitness functions, etc [35–37]. However, for our research, we select the extensively validated NSGA-II [13], due to its established efficiency [38] and widespread adoption in the engineering field [39]. It is important to highlight that while multi-objective GAs can find optimal solutions in the *extended* design

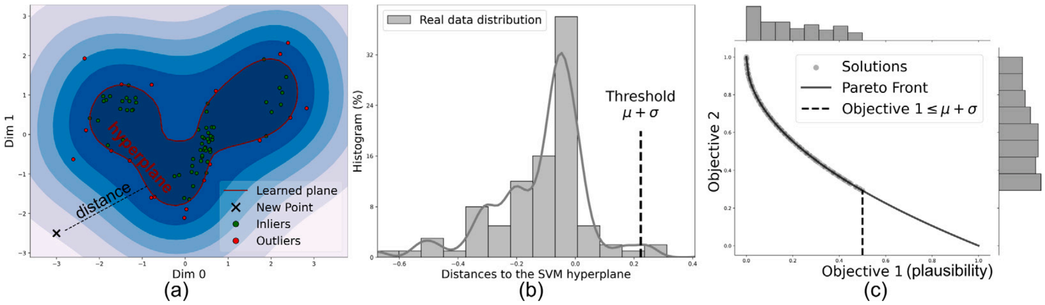


Fig. 6. Graphical representation of the plausibility objective and threshold calculation for the one-class SVM. **(a)** Depicts the learned SVM hyperplane. **(b)** Shows the distribution of the signed distances of real data points to the hyperplane and the chosen threshold. **(c)** Represents example Pareto-optimal GA-generated solutions constrained by the threshold.

space, these extensions and all objectives must be well-defined. Consequently, GAs are confined to exploring and generating solely within the boundaries of their predefined search space.

NSGA-II: To solve the MOO problem, we deploy the Non-Dominated Sorting Genetic Algorithm (NSGA-II) [13]. Initially, a random initial population is created within the bounds of the *extended design space* (Section 2.2). All candidates are evaluated by the objectives and ranked into dominating fronts based on the non-dominated sorting algorithm. To maintain diversity in the objective space, fronts are sorted based on their crowding distance. Top-ranking individuals are promoted to the *reproduction/selection* phase, where crossover and mutation are performed to create the population of the next generation. The surviving individuals of the final generation are gathered and returned as the optimal trade-off solutions to the problem.

In summary, our related work covers all methods used in our study, explaining how they operate in great detail. We investigate data-driven methods for mixed tabular data generation, and multi-objective-driven methods based on GAs for generating data outside the existing data manifold.

4. Integrated approach for mixed tabular data generation under data scarcity and imbalance

Our objective is to improve the accuracy and generalization capabilities of ML models on a downstream task (e.g., predicting the cost of a structure) by augmenting the training data through synthetic jacket designs. This task is particularly demanding due to challenges such as limited data, imbalances, and mixed tabular data, which are common issues in the engineering field. As mentioned earlier, relying solely on a data-driven approach has limitations, as the effectiveness of such methods is closely linked to the quality and representativeness of the training data. Consequently, these methods may struggle to address the significant imbalances inherent in the original data, as illustrated in Fig. 2. Recognizing this, we also incorporate more traditional AI methods that search for good designs in the extended design space (Section 2.2).

4.1. Data-driven generation

To learn the distribution of the data, we compare three data-driven methods tailored for mixed tabular data generation (analytically described in Section 3.1), namely:

- MDGMM: a probabilistic model relying on Gaussian mixtures [12]
- TVAE: a variational autoencoder neural network [11]
- LLM: a large language model-based approach [1]

4.2. Multi-objective design space exploration under plausibility constraints

To further enable design generation beyond the limitations of the training data, we employ a multi-objective genetic algorithm, NSGA-II, that can find optimal jacket designs in our extended design space (Section 2.2), introducing more variety to the generated data, for example, in terms of brace types C.

The design space consists of the features described in Table 1 and the extended value domain. The evaluation space consists of cost and load-bearing capacity objectives, as discussed in Table 2. Nevertheless, our aim is for the solutions generated by the GA to be realistic. To achieve this, we incorporate a *plausibility objective* and *plausibility threshold* into the problem, controlling the extent to which the GA can explore and generate solutions that deviate from the distribution of real data. We define plausibility as the distance to the real data manifold and denote it as the *plausibility metric*, and strive to minimize it. Measuring the distance of an instance to the real data manifold becomes particularly difficult, especially under limited data availability. Moreover, the presence of categorical variables poses an additional challenge, as employing one-hot encoding can result in higher-dimensional inputs (compared to the number of real observations). In this study, we cover and compare three different outlier-detection methods that are well-suited for our scarce tabular dataset:

- **Isolation Forest:** Isolates outliers by constructing binary trees [40]. It assigns anomaly scores based on the average path lengths of data points in these trees, with shorter paths indicating higher anomaly likelihood. The tree-like structure of the algorithm is especially efficient in high-dimensional datasets.
- **One-class SVM:** One-class Support Vector Machines (SVM) [41] learn a hyperplane to encapsulate “normal” instances. SVMs work especially well on small, high-dimensional datasets, such as ours. The objective is to minimize the distance of samples to the learned hyperplane, bringing instances closer to the real data.
- **KDE:** Kernel Density Estimation (KDE) [42], estimates the probability density of the dataset. Samples with a larger log-likelihood under the model, are more likely to belong to the real data.

However, relying solely on the plausibility objective does not ensure that all solutions will align close to the data manifold, since NSGA-II finds trade-off solutions across the objective space. For example, some solutions might be less plausible, but perform better in terms of other objectives, like cost, etc. To effectively confine the generated designs in relation to their similarity to the real data, we propose establishing a *plausibility threshold* for the optimization problem. To enable a consistent comparison of threshold values among various plausibility methods, we set it dynamically based on the distribution of the plausibility metrics observed in the real data.

In Fig. 6 we present an illustrative example for determining the plausibility threshold based on a one-class SVM. First, the SVM is fitted on the real instances of our dataset (scattered points), learning a hyperplane that encapsulates most real instances (green points). The slack parameter in SVMs allows some misclassification (red points) while achieving a smoother decision boundary **(a).** To set a plausibility threshold, we first calculate the median and standard deviation (μ, σ) of the plausibility metric for all real instances. In the case of SVMs, we compute the signed distances of all real instances to the decision

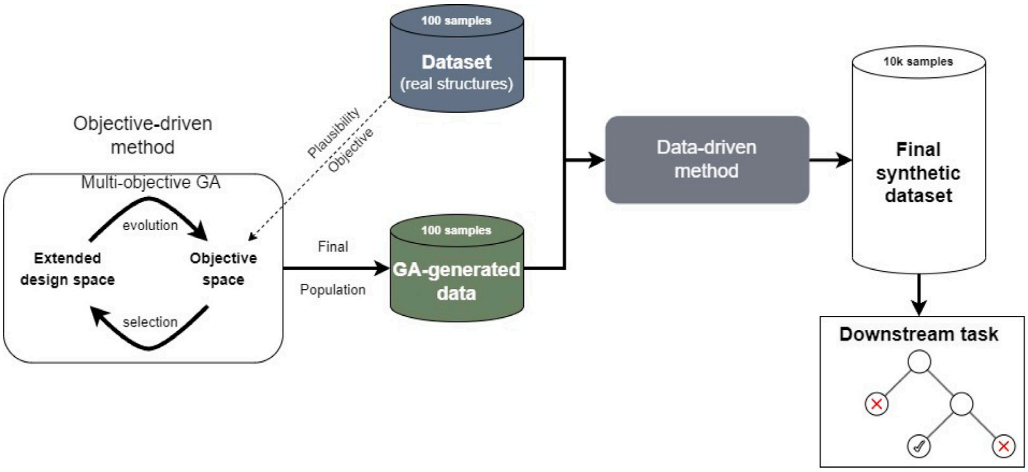


Fig. 7. Integrated approach for generating synthetic designs, combining objective-driven and data-driven methods.

boundary **(b)**. Subsequently, we choose a threshold in the form of $thr = \mu + k \cdot \sigma$, where k dynamically controls the impact of the threshold, with smaller values enforcing a stricter criterion. For example, setting k=0 imposes a strict threshold, only accepting solutions with a plausibility metric lower than the median value of the real data. This means that only instances with a plausibility distance smaller than 50% of the real data are considered plausible. When increasing the threshold by adding one or more standard deviations, a larger portion of the data is classified as inliers, thus permitting greater variation in solutions produced by the GA. The threshold value is set as an inequality constraint for the plausibility objective of the optimization problem ($obj \leq thr$), constraining the Pareto-optimal solutions of the objective space, as depicted in subfigure **(c)**.

Although our evolutionary approach uses multiple objectives (listed in Table 2), we only apply a threshold to the plausibility objective. Other constraints, such as limiting the maximum cost of a structure, are not yet incorporated, however, it is feasible to integrate similar constraints effortlessly into our methodology.

In our experiments, we show that the dynamic nature of the threshold directly influences the plausibility and diversity trade-off, and this effect is consistent across all employed plausibility methods.

4.3. Integrated approach: combining objective-driven and data-driven modeling

Both solely data-driven or objective-driven approaches have their limitations for improving predictive model performance and generalization. Data-driven methods can be used for improving predictions on the dataset domain. On the other hand, objective-driven methods will introduce more diversity in the training data, aiding models to generalize in the extended domain, albeit at the cost of reduced performance on the original real data. Additionally, objective-driven methods are much more computationally expensive, since they require structural simulation. This becomes computationally unfeasible, especially for large populations. Thus, we maintain the population size at 100 samples, matching the size of the real dataset.

To overcome these problems, we propose to combine both methods sequentially, as shown in Fig. 7. First, the multi-objective GA is employed to find optimal, novel, but plausible designs by leveraging the extended space. The synthetic population generated by the GA is concatenated with the real dataset to train the data-driven method on samples from both domains. The data-driven method is then used to efficiently generate the final large population of $10k$ samples, which is used as training data for the downstream task. Our integrated generative method combines data-driven and objective-driven approaches to expand the coverage of the existing design space, enhancing both the quality and diversity of synthetic data. While our approach shares similarities with GANs, utilizing two models, it differs fundamentally in how new designs are generated. Traditional GANs learn exclusively from available data, limiting their ability to incorporate additional information, such as simulation results, to overcome existing imbalances in the data. In contrast, our method employs a genetic algorithm to explore a broader design space, optimizing jacket designs based on simulation outcomes. A data-driven model then learns the statistical distribution of both real and GA-generated designs, synthesizing additional plausible samples that go beyond what GANs alone could achieve.

In our experiments, we observe that the synthetic training data generated by our integrated approach enhances performance when tested on structures from both the original and the extended design space.

4.4. Evaluating synthetic data: predictive performance and diversity

As our goal is to generate synthetic data to improve ML performance and generalization, we use ML efficiency, i.e., testing a model's performance on a downstream task, as our primary evaluation metric. Additionally, we assess the diversity in synthetic data using an entropy metric for both discrete and continuous variables, which is explained hereafter.

ML efficiency on the downstream task: We construct a downstream task to test synthetic data quality. Since our dataset does not contain specific target labels, we formulate the following regression problem. The inputs are feature-vector representations of jacket structures defined in Table 1, and the outputs (targets) are all objectives (cost and load-bearing capacity) of Table 2, which are calculated for both real and synthetic structures. This is a very common predictive task in offshore engineering referred to as surrogate modeling [43–45], where a predictive model is used to substitute an expensive analytical or simulated computation. Efficiency is measured by typical metrics for regression problems that measure the difference between the true outputs, compared to the predicted values, i.e. Mean Squared Error (MSE) and "R-squared" R^2. We use four models for Machine Learning efficiency (MLE), namely Random Forest (RF) and Decision Tree (DT). Additionally, we integrate XGBoost [46], a well-established gradient boosting method recognized for its high accuracy, efficient optimization, and scalability. Most importantly, similar to [47], we adopt the CatBoost model [48], which is the leading gradient boosting DT approach, with state-of-the-art performance for mixed tabular data [49].

Data entropy: To assess the diversity of the synthetic samples we measure the total entropy in the data. High entropy signifies greater

Table 3
Comparison of all generative methods based on supported features, number of trainable parameters, and runtime.

		MDGMM	TVAE	LLM	NSGA-II
Features	Mixed	✓	✓	✓	✓
	Ordinal/ Nominal	✓	✗	✗	✗
	Names	✗	✗	✓	✗
#Trainable parameters		57	$46k$	$82M$ (pre-trained)	–
Runtime (s)	Train (100)	1.14	1.27	105	–
	Sample (100)	0.05	0.11	12	387
	Sample (500)	0.08	0.12	57	1150
	Sample (1k)	0.13	0.15	110	2311

variety, while low entropy suggests more uniformity. Since our data contains both discrete and continuous variables, we report the *discrete entropy* and *differential entropy* respectively.

$$H(X) = \begin{cases} -\sum_i p(x_i)\log p(x_i) & \text{for discrete } X \\ -\int_{-\infty}^{\infty} f(x)\log f(x)\,dx & \text{for continuous } X \end{cases} \tag{3}$$

where H is the entropy, X is the input data, and f is the probability density function (for continuous features).

5. Experimental evaluation

In this section, we present the findings of our research. Initially, we assess data-driven approaches to identify the most effective one for our data specifications. Subsequently, we analyze the outcomes generated by the multi-objective genetic algorithm. Lastly, we evaluate our proposed solution that integrates both data-driven and objective-driven methods.

First, we evaluate all generative methods by considering their fundamental algorithmic characteristics and computational costs (Table 3). In terms of feature types, all methods inherently support mixed features, encompassing both discrete and continuous features in coordination with the data domain of our work. Nonetheless, in the case of MDGMM, discrete data types are explicitly addressed, distinguishing between ordinal, nominal, binary, etc., categorical types. Specifically, we model the number of layers (L, ordinal) with an ordered multinomial distribution, the number of legs (N, binary) with a Bernoulli, and the brace types (C, nominal) with an unordered multinomial distribution. In contrast, all other approaches amalgamate these into a single categorical data type. Moreover, LLM is the only approach where the names of the features can influence the data generation process. While a longer, more detailed description of each feature could provide additional information to the language model, longer tokens also increase computational time during sampling.

Furthermore, we present the number of trainable (learnable) parameters, which determines model complexity. More complex models have the advantage of representing more complicated functions, but are prone to overfitting. This is especially relevant for our application because we work under data scarcity, where overfitting is a critical concern.

Finally, we perform a runtime comparison. For data-driven methods, we also account for the training time, which is generally low for all approaches, due to the limited amount of data. However, it is worth noting that fine-tuning the LLM can be a resource-intensive process, taking, for instance, more than 9 hours for larger datasets exceeding $10k$ samples [1]. More importantly, the objective-driven NSGA-II exhibits significantly higher computational complexity, dependent on both the population size and number of generations, and notably, the computation of all objectives. This becomes even more substantial for simulated objectives since most finite element method (FEM) simulators (see Section 2.4) do not support GPU parallelization. Furthermore, while the load cases taken into account in our study are relatively simple, more intricate or high-fidelity simulations (e.g. fatigue analysis [34]) could render the computational time impractical for larger populations without the integration of surrogate modeling.

Based on the information presented in Table 3, we conclude that MDGMM stands out as the most effective approach for generating synthetic jacket designs, particularly in our scenario featuring a mixed tabular representation and a limited number of training instances. This conclusion is supported by both qualitative and quantitative results presented below.

5.1. Experimental setup

To produce robust results, we conduct the synthetic generation process five times using different random initialization seeds. Additionally, we repeat each MLE experiment five times, each time executing three-fold cross-validation, using default hyperparameters for all downstream models. Experimental results are reported as the average values and standard deviations, aggregated over all random seed initializations for the synthetic data generation and all cross-validations.

5.2. Data-driven methods

In this assessment, we examine all data-driven methods, to determine the most effective in capturing the real-data distribution of our small dataset of real jacket designs.

5.2.1. Qualitative analysis

We start by qualitatively observing the feature distributions of the real data, compared to the synthetic data in Fig. 8. For continuous features (top row), we observe that MDGMM learns smoother distributions, whereas the LLM tends to oversample the most frequent feature values (e.g., structures with a total height of around 50, or structures with 4 layers). Striving to precisely capture the fluctuations in the training data may not accurately reflect the underlying patterns and could result in poorer performance during testing when the distribution may vary slightly. For discrete features (bottom row) MDGMM learns the real-data distribution more consistently than LLM or TVAE. This is due to the advantageous explicit modeling of the discrete data types performed in MDGMM.

To summarize, by looking at the per-feature distributions, MDGMM stands out as the best data-driven method for capturing the real data distribution.

5.2.2. Quantitative analysis (MLE)

Although the qualitative analysis is a good first indication, it is limited to per-feature similarity and does not account for feature interdependencies. As mentioned previously, to quantitatively assess synthetic data quality, we measure the ML efficiency on a downstream task for three predictive models. For this experiment, the real designs of our dataset are split into a $1/3$ test set, and $2/3$ is used for training. Keeping the same real test set, we train using $10k$ synthetic instances of each generative method. Results are presented in Table 4.

In Table 4, we observe that all synthetic data generation methods lead to improved performance for all ML models. This improvement can

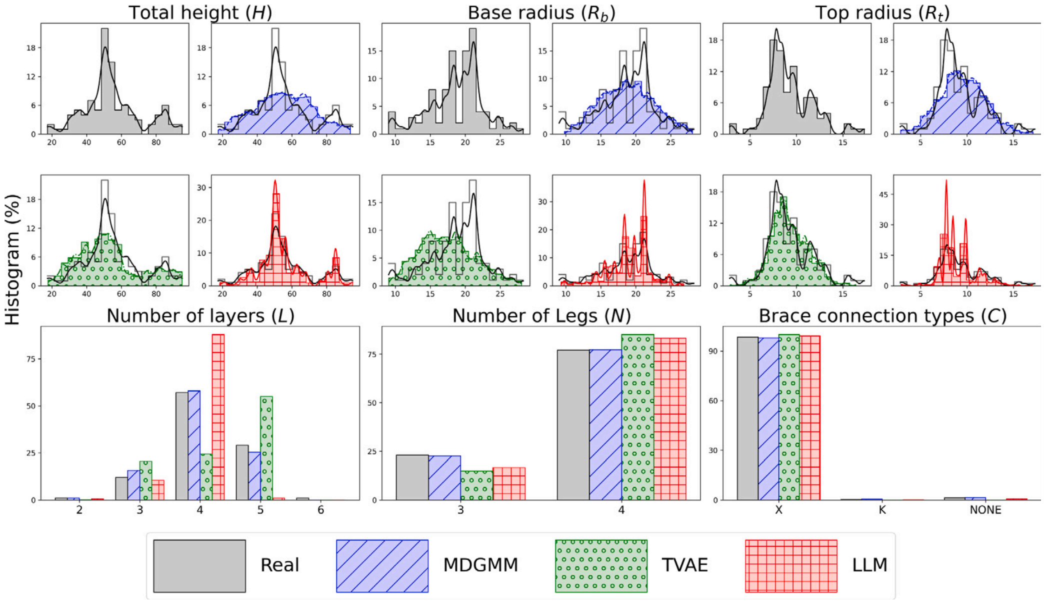


Fig. 8. Histogram distributions for continuous and discrete features of real data, and synthetic data generated by data-driven methods.

Table 4
MLE results on all predictive models, for all data-driven methods.

Model	Train data	MSE ↓	R^2 ↑
CatBoost	Real	0.018 (0.007)	0.311 (0.078)
	MDGMM	**0.009** (0.003)	**0.582** (0.106)
	TVAE	0.013 (0.005)	0.537 (0.063)
	LLM	0.010 (0.004)	0.569 (0.072)
DT	Real	0.012 (0.005)	0.479 (0.348)
	MDGMM	**0.003** (0.001)	**0.861** (0.051)
	TVAE	0.004 (0.001)	0.825 (0.038)
	LLM	0.005 (0.003)	0.787 (0.069)
RF	Real	0.008 (0.004)	0.700 (0.096)
	MDGMM	**0.001** (0.001)	**0.949** (0.020)
	TVAE	0.002 (0.001)	0.910 (0.020)
	LLM	0.003 (0.002)	0.878 (0.043)
XGBoost	Real	0.009 (0.005)	0.574 (0.414)
	MDGMM	**0.001** (0.000)	**0.970** (0.013)
	TVAE	0.001 (0.001)	0.949 (0.013)
	LLM	0.001 (0.000)	0.954 (0.015)

be attributed to the fact that ground-truth targets for the downstream task are not generated (synthetic), but rather reliably computed, for instance, through simulation (refer to Section 4.4). However, it is important to note that similar improvements are generally not expected in applications where the labels have to be generated along with the synthetic samples [1,11].

As expected, MDGMM outperforms all other approaches for all evaluation metrics and target models. For the CatBoost model, training with synthetic samples generated by MDGMMincreases predictive performance by +45% for both evaluation metrics on the real test data. Notably, we observe that training XGBoost on synthetic samples generated by MDGMM, greatly improves performance, increasing the R^2 score by +70% when compared to training on real data.

Moreover, we perform an additional experiment to investigate how varying the magnitude of synthetic data, influences the improvement of model accuracy. Experimental iterations ranging from 100 to 40k synthetic samples are conducted, with increments of 100 samples in each step. In Fig. 9, we observe that MDGMM outperforms the other two generative approaches, while performance is saturated for all methods after 10k synthetic samples.

5.3. Objective-driven method

Although data-driven methods are effective in generating jacket designs that closely resemble the real data distribution, they struggle to address imbalances inherent in the actual dataset (see Fig. 8). Moreover, these methods are confined to the design space dictated by the real data. To overcome these issues, we use a multi-objective GA, NSGA-II, to explore optimal solutions in our extended design space (outlined in Section 2.2). To regulate how far the GA can deviate from the real data manifold during generation, we introduced a plausibility objective and threshold.

Fig. 10 illustrates an example distribution of brace types used in solutions discovered by NSGA-II with and without the plausibility constraint. It is evident that the plausibility-constrained GA (yellow) generates designs that closely align with real designs but does not explore many alternative options.

To quantify the effect of the plausibility constraint on the generated synthetic data, we conduct the following experiment. We gradually increase the k parameter of the plausibility threshold (see Fig. 6) to create a range of threshold values $thr_{range} = (\mu, \mu + 4 \cdot \sigma)$. For each set of generated solutions, we evaluate on MLE, as well as our entropy metrics, which quantify data variety. In Fig. 11, we observe a deterioration in MLE as the threshold is increased, reflecting that more out-of-distribution solutions are generated. However, this also leads to an increase in the entropy of the data, indicating that novel designs are created.

Notably, the scatter plots depicted in Fig. 11 exhibit some inconsistency, for example, there are instances where an increase in threshold values results in a minor decrease in MLE. This inconsistency can be attributed to the calculation of the threshold values (see Section 4.2), which is subject to noise, particularly given our small dataset. Nevertheless, the objective of this experiment is to demonstrate the overall trend of increased diversity and MSE with higher threshold values.

This intended effect of the plausibility threshold is observed for all three plausibility methods (defined in Section 4) and showcases the conflicting nature of the MLE and the entropy metrics, as well as, the effectiveness of our dynamic plausibility threshold.

5.4. Integrated data-driven and objective-driven method

In our previous experiments, MDGMM was identified as the most suitable method for synthesizing jacket designs closely resembling

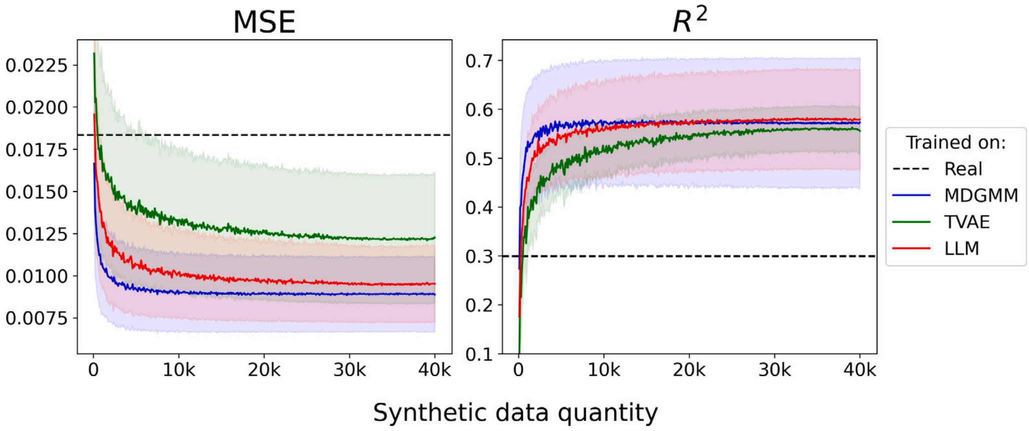


Fig. 9. Improvement of MLE metrics (CatBoost model) due to increased number of synthetic samples.

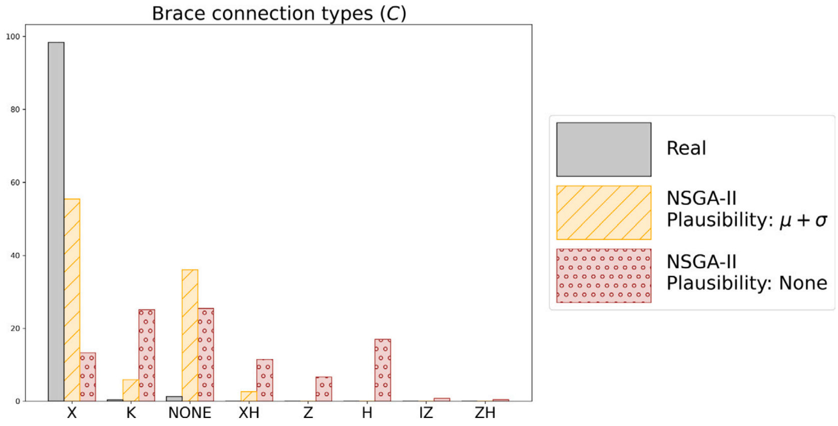


Fig. 10. Histogram distributions of the brace type feature (C) for real and NSGA-II generated data. Plausibility-constrained solutions (yellow) align more closely with the real data, while non-constrained solutions (red) explore alternatives more extensively.

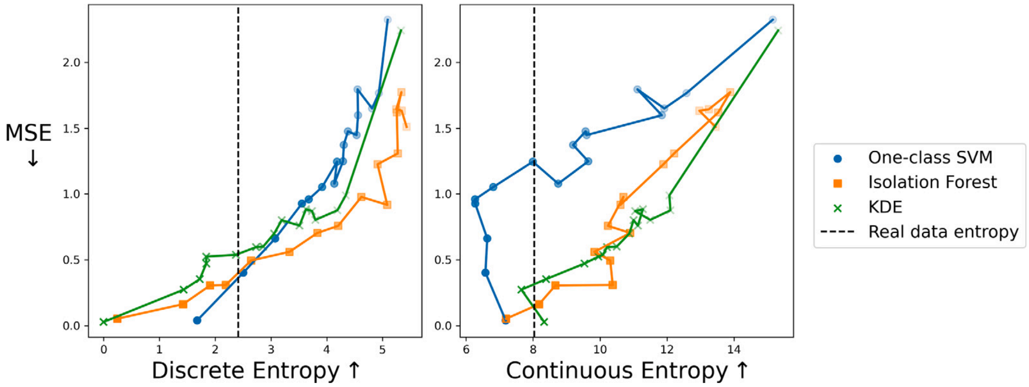


Fig. 11. Joint scatter plot of evaluation metrics for different runs of NSGA-II, by gradually increasing (left → right) the plausibility threshold.

those in our dataset. However, to generate more novel designs, we employed the objective-driven NSGA-II. In this experiment, we evaluate MLE by training and testing models on both real data and data generated by NSGA-II (each comprising 100 instances).

In Table 5, we observe the expected: (i) models exclusively trained on real data demonstrate strong performance on previously unseen real data but show limitations when faced with “out-of-distribution” NSGA-II data from the extended design space, (ii) training on NSGA-II and evaluating on real data results in performance degradation, as models struggle to generalize on a different testing distribution, and (iii) training on a combined set of real and NSGA-II data leads to some improvement for both test sets since the training set contains samples from both distributions.

Nonetheless, to further enhance results, we deploy our integrated approach. That is, we use MDGMM to generate $10k$ synthetic instances from the combined training set (real, NSGA-II). This approach consistently yields the best results, significantly aiding model generalization on novel NSGA-II generated data while maintaining high performance on real jacket designs. Remarkably, for the DT, RF, and XGB models, our integrated approach even improves the MLE on both test sets, while XGB achieves the best overall improvement. For instance, training on the $10k$ synthetic samples generated by our integrated approach leads

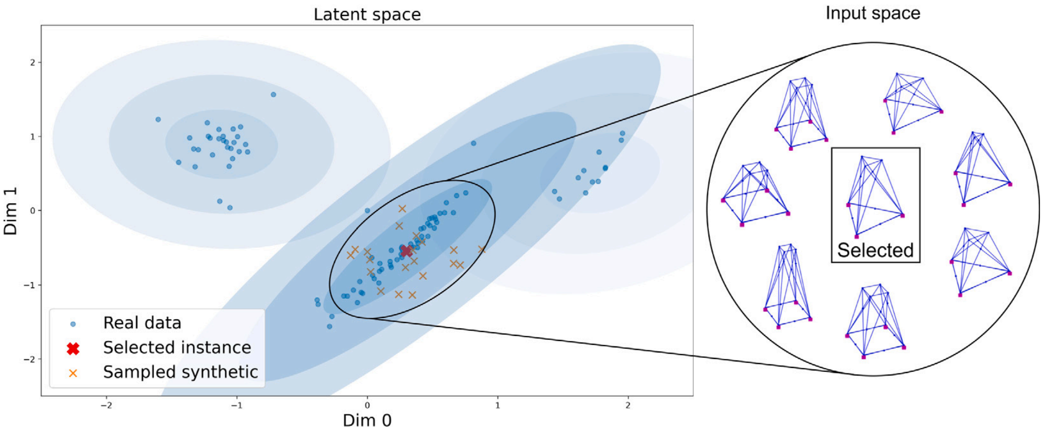


Fig. 12. Example application for sampling synthetic design alternatives from the latent space.

Table 5
MLE results on all predictive models, testing all methods on both real and NSGA-II generated data.

Model	Train data	Test data			
		Real		NSGA-II	
		MSE ↓	R^2 ↑	MSE ↓	R^2 ↑
CBR	Real	**0.018** (0.007)	**0.310** (0.077)	6.703 (0.846)	−0.691 (0.183)
	NSGA-II	2.196 (0.322)	−128.845 (58.474)	1.794 (0.175)	0.465 (0.045)
	Real, NSGA-II	0.195 (0.035)	−10.167 (4.577)	1.860 (0.187)	**0.445** (0.027)
	MDGMM(Real, NSGA-II)	0.150 (0.004)	−12.231 (7.697)	**1.857** (0.132)	0.290 (0.022)
DT	Real	0.012 (0.005)	0.479 (0.348)	6.208 (0.797)	−0.493 (0.168)
	NSGA-II	2.672 (1.102)	−154.388 (81.054)	1.229 (0.455)	0.630 (0.122)
	Real, NSGA-II	0.167 (0.191)	−4.972 (6.954)	1.243 (0.506)	0.631 (0.111)
	MDGMM(Real, NSGA-II)	**0.006** (0.001)	**0.670** (0.130)	**0.551** (0.132)	**0.704** (0.066)
RF	Real	0.008 (0.004)	0.699 (0.093)	6.232 (0.816)	−0.464 (0.126)
	NSGA-II	1.368 (0.426)	−81.644 (40.761)	0.616 (0.217)	0.802 (0.072)
	Real, NSGA-II	0.072 (0.063)	−1.617 (1.786)	0.611 (0.248)	**0.812** (0.068)
	MDGMM(Real, NSGA-II)	**0.002** (0.002)	**0.870** (0.070)	**0.541** (0.119)	0.754 (0.038)
XGB	Real	0.009 (0.005)	0.572 (0.414)	5.438 (0.621)	−0.286 (0.144)
	NSGA-II	0.352 (0.213)	−14.074 (9.645)	0.285 (0.065)	0.858 (0.045)
	Real,NSGA-II	0.027 (0.033)	0.094 (0.993)	0.297 (0.132)	0.858 (0.056)
	MDGMM(Real,NSGA-II)	**0.002** (0.001)	**0.927** (0.015)	**0.092** (0.024)	**0.956** (0.013)

to improved performance when tested on real data, surpassing the results obtained by training solely on real data.

Similar to our previous experiments, we perform 3-fold cross-validation (70%/30% train/test data) for the real and NSGA-II data, each comprised of 100 instances, while the MDGMM generated data (integrated approach) consists of $10k$ training samples.

6. Discussion and potential applications

In this work, we tackle the problem of synthetic tabular data generation under data scarcity. This problem setting is especially valid for engineering problems since engineers often use this modality for saving data and design specifications, e.g., in databases, forms, configuration files, etc. Real-world applications of our generative approach can range from sampling strategies for the initialization of optimization problems to improving the robustness and accuracy of surrogate models.

The second problem that we face is generating outside of the existing data-manifold, and out-of-distribution generalization. We generate using the objective-driven NSGA-II and we enhance predictive performance by populating a synthetic dataset using the data-driven MDGMM. A relevant possible application from the offshore wind domain could be the generation of novel substructures of the future. Specifically, as wind turbine blades grow to increase power production, so will the offshore substructures, possibly leading ML models to generalize on a previously unexplored setting with limited prior knowledge.

Finally, models that learn a lower-dimensional latent representation of the real data like MDGMM and TVAE, can be used for many different applications that are difficult to perform in the tabular input space, such as visualization, clustering, and measuring distance or similarity. Another useful property of the latent space for engineers is the generation of synthetic design alternatives by sampling in the “neighborhood” of an existing design.

Fig. 12 illustrates this concept, demonstrating its particular utility in the initial conceptual design phase, where it can serve to inspire and rapidly explore novel design possibilities.

7. Conclusions

This paper addressed the important problem of data scarcity and imbalances, often observed in the engineering domain, by combining data-driven and objective-driven methods for synthesizing mixed tabular data. Three distinct generative AI approaches were evaluated, highlighting the efficacy of the Mixed Deep Gaussian Mixture Model (MDGMM) in generating synthetic instances that closely mimic real offshore jacket designs. This lightweight probabilistic model proves particularly effective for augmenting small datasets, showcasing its potential for addressing data challenges in engineering applications.

Additionally, a multi-objective Genetic Algorithm (GA) that optimizes offshore jacket designs was proposed by exploring trade-off solutions in a domain-specific objective space. We direct the generation process towards the real data manifold by introducing a plausibility objective and studying its effect on the diversity of the generated offshore jacket designs. Our integrated approach combines data-driven and objective-driven methodologies, fulfilling our goal of increased model performance and generalization capabilities on a downstream task. Our research not only advances generative AI applications in engineering but also provides a versatile framework for addressing data limitations across various domains.

Nonetheless, there are limitations in our work that suggest opportunities for future enhancements. Specifically, we primarily concentrate on the initial conceptual design phase, but there is potential for expanding the objective space to encompass additional stages in the lifecycle of a structure. Furthermore, improved dynamic structural objectives can be utilized by adjusting the load cases based on e.g., the size of a structure. Likewise, various structural tests can be performed accounting e.g., for fatigue damage or overlapping eigenfrequencies. Furthermore, while our integrated approach learns from real-world data and simulations, integrating engineering expertise into the design and evaluation of structures would be beneficial. This can be achieved through an expert-driven human-in-the-loop process that captures the intuition and knowledge of engineers or by leveraging expert feedback on existing structures. However, these methods are more resource-intensive compared to purely data-driven approaches. Finally, we would like to perform conditional data-driven generation in the future, leveraging site-specific information for task-specific applications.

CRediT authorship contribution statement

Emmanouil Panagiotou: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation. **Han Qian:** Writing – review & editing, Validation, Methodology, Investigation, Data curation, Conceptualization. **Steffen Marx:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization. **Eirini Ntoutsi:** Writing – review & editing, Validation, Supervision, Project administration, Investigation, Funding acquisition, Conceptualization.

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.

Acknowledgments

This study is funded by the German Research Foundation (DFG) as part of the Collaborative Research Center 1463 (CRC1463) “Integrated Design and Operation Methodology for Offshore Megastructures” (project number 434502799). The authors gratefully acknowledge the financial support through the DFG.

Data availability

Data will be made available on request.

References

[1] V. Borisov, K. Sessler, T. Leemann, M. Pawelczyk, G. Kasneci, Language models are realistic tabular data generators, in: The Eleventh International Conference on Learning Representations, 2023, URL openreview.net/forum?id=cEygmQNOeI. (Last Access August 10 2024).

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 1877–1901, URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[3] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, M. Chen, Hierarchical text-conditional image generation with clip latents, 2022, http://dx.doi.org/10.48550/arXiv.2204.06125, arXiv preprint arXiv:2204.06125 1 (2) 3.

[4] 4C Offshore Ltd., 4C offshore construction and maintenance vessel online database (2014), 2024, URL 4coffshore.com/windfarms/default.aspx. (Last Access August 10 2024).

[5] S.M. Harle, Advancements and challenges in the application of artificial intelligence in civil engineering: a comprehensive review, Asian J. Civ. Eng. (2023) 1–18, http://dx.doi.org/10.1007/s42107-023-00760-9.

[6] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, G. Kasneci, Deep neural networks and tabular data: A survey, IEEE Trans. Neural Netw. Learn. Syst. (2022) http://dx.doi.org/10.1109/TNNLS.2022.3229161.

[7] F. Luleci, F.N. Catbas, A brief introductory review to deep generative models for civil structural health monitoring, AI Civ. Eng. 2 (1) (2023) 9, http://dx.doi.org/10.1007/s43503-023-00017-z.

[8] W. Liao, X. Lu, Y. Fei, Y. Gu, Y. Huang, Generative AI design for building structures, Autom. Constr. 157 (2024) 105187, http://dx.doi.org/10.1016/j.autcon.2023.105187.

[9] A.N. Wu, R. Stouffs, F. Biljecki, Generative adversarial networks in the built environment: A comprehensive review of the application of GANs across data types and scales, Build. Environ. 223 (2022) 109477, http://dx.doi.org/10.1016/j.buildenv.2022.109477.

[10] H. Qian, E. Panagiotou, S. Marx, E. Ntoutsi, Data-based conceptual design of offshore jackets using a self-developed database, in: ISOPE International Ocean and Polar Engineering Conference, ISOPE, 2023, pp. ISOPE–I, URL onepetro.org/ISOPEIOPEC/proceedings-abstract/ISOPE23/All-ISOPE23/524556. (Last Access August 10 2024.

[11] L. Xu, M. Skoularidou, A. Cuesta-Infante, K. Veeramachaneni, Modeling tabular data using conditional gan, Adv. Neural Inf. Process. Syst. 32 (2019) http://dx.doi.org/10.48550/arXiv.1907.00503.

[12] R. Fuchs, D. Pommeret, C. Viroli, Mixed deep Gaussian mixture model: a clustering model for mixed datasets, Adv. Data Anal. Classif. 16 (1) (2022) 31–53, http://dx.doi.org/10.48550/arXiv.2010.06661.

[13] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197, http://dx.doi.org/10.1109/4235.996017.

[14] E. Panagiotou, H. Qian, M. Wynants, A. Kriese, S. Marx, E. Ntoutsi, Explainable AI-based generation of offshore substructure designs, in: ISOPE International Ocean and Polar Engineering Conference, ISOPE, 2023, pp. ISOPE–I, URL onepetro.org/ISOPEIOPEC/proceedings-abstract/ISOPE23/All-ISOPE23/524270. (Last Access August 10 2024).

[15] I.-W. Chen, B.-L. Wong, Y.-H. Lin, S.-W. Chau, H.-H. Huang, Design and analysis of jacket substructures for offshore wind turbines, Energies 9 (4) (2016) 264, http://dx.doi.org/10.3390/en9040264.

[16] X. Han, A. Chen, B. Zhou, G. Zhang, W.M. Gho, Strength performance of an eccentric jacket substructure, J. Mar. Sci. Eng. 7 (8) (2019) 264, http://dx.doi.org/10.3390/jmse7080264.

[17] A. Panesar, M. Abdi, D. Hickman, I. Ashcroft, Strategies for functionally graded lattice structures derived using topology optimisation for additive manufacturing, Addit. Manuf. 19 (2018) 81–94, http://dx.doi.org/10.1016/j.addma.2017.11.008.

[18] N.K. Vemula, W. De Vries, T. Fischer, A. Cordle, B. Schmidt, Design solution for the upwind reference offshore support structure (2010), 2024, URL ewea.org/fileadmin/files/library/publications/reports/UpWind_Report.pdf. (Last Access August 10 2024).

[19] J. Jonkman, S. Butterfield, W. Musial, G. Scott, Definition of a 5-MW Reference Wind Turbine for Offshore System Development, Tech. Rep., National Renewable Energy Lab.(NREL), Golden, CO (United States, 2009, http://dx.doi.org/10.2172/947422.

[20] J. Morison, J. Johnson, S. Schaaf, The force exerted by surface waves on piles, J. Pet. Technol. 2 (05) (1950) 149–154, http://dx.doi.org/10.2118/950149-G.

[21] H. Lian, A.N. Christiansen, D.A. Tortorelli, O. Sigmund, N. Aage, Combined shape and topology optimization for minimization of maximal von mises stress, Struct. Multidiscip. Optim. 55 (5) (2017) 1541–1557, http://dx.doi.org/10.1007/s00158-017-1656-x.

[22] J. Häfele, R.R. Damiani, R.N. King, C.G. Gebhardt, R. Rolfes, A systematic approach to offshore wind turbine jacket predesign and optimization: geometry, cost, and surrogate structural code check models, Wind. Energy Sci. 3 (2) (2018) 553–572, http://dx.doi.org/10.5194/wes-3-553-2018.

[23] M. Zhu, F. McKenna, M.H. Scott, OpenSeesPy: Python library for the OpenSees finite element framework, SoftwareX 7 (2018) 6–11, URL openseespydoc.readthedocs.io. (Last Access August 10 2024).

[24] B. Asgarian, M. Zarrin, M. Sabzeghabaian, Effect of foundation behaviour on steel jacket offshore platform failure modes under wave loading, Ships Offshore Struct. 14 (6) (2019) 570–581, http://dx.doi.org/10.1080/17445302.2018.1526862.

[25] D.P. Kingma, M. Welling, Auto-encoding variational bayes, 2013, http://dx.doi.org/10.48550/arXiv.1312.6114, arXiv preprint arXiv:1312.6114.

[26] C.M. Bishop, N.M. Nasrabadi, Pattern Recognition and Machine Learning, vol. 4, Springer, ISBN: 978-0-387-31073-2, 2006.

[27] Z. Ghahramani, G.E. Hinton, The EM Algorithm for Mixtures of Factor Analyzers (1996), Tech. Rep., Technical Report CRG-TR-96-1, University of Toronto, URL ecs.toronto.edu/~hinton/absps/tr-96-1.pdf. (Last Access August 10 2024).

[28] I. Moustaki, M. Knott, Generalized latent trait models, Psychometrika 65 (2000) 391–411, http://dx.doi.org/10.1007/BF02296153.

[29] S. Cagnone, C. Viroli, A factor mixture model for analyzing heterogeneity and cognitive structure of dementia, AStA Adv. Stat. Anal. 98 (2014) 1–20, http://dx.doi.org/10.1007/s10182-012-0206-5.

[30] R. Fuchs, D. Pommeret, S. Stocksieker, MIAMI: Mixed data augmentation mixture, in: International Conference on Computational Science and Its Applications, Springer, 2022, pp. 113–129, http://dx.doi.org/10.1007/978-3-031-10522-7_9.

[31] K. Lv, Y. Yang, T. Liu, Q. Gao, Q. Guo, X. Qiu, Full parameter fine-tuning for large language models with limited resources, 2023, http://dx.doi.org/10.48550/arXiv.2306.09782, arXiv preprint arXiv:2306.09782.

[32] E.J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, 2021, http://dx.doi.org/10.48550/arXiv.2106.09685, arXiv preprint arXiv:2106.09685.

[33] K.-H. Chew, K. Tai, E. Ng, M. Muskulus, Analytical gradient-based optimization of offshore wind turbine substructures under fatigue and extreme loads, Mar. Struct. 47 (2016) 23–41, http://dx.doi.org/10.1016/j.marstruc.2016.03.002.

[34] A.A. Motlagh, N. Shabakhty, A. Kaveh, Design optimization of jacket offshore platform considering fatigue damage using genetic algorithm, Ocean Eng. 227 (2021) 108869, http://dx.doi.org/10.1016/j.oceaneng.2021.108869.

[35] T. Murata, H. Ishibuchi, MOGA: multi-objective genetic algorithms, in: IEEE International Conference on Evolutionary Computation, Vol. 1, IEEE Piscataway, 1995, pp. 289–294, http://dx.doi.org/10.1109/ICEC.1995.489161.

[36] H. Lu, G.G. Yen, Rank-density based multiobjective genetic algorithm, in: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600), Vol. 1, IEEE, 2002, pp. 944–949, http://dx.doi.org/10.1109/CEC.2002.1007052.

[37] X. Zou, Y. Chen, M. Liu, L. Kang, A new evolutionary algorithm for solving many-objective optimization problems, IEEE Trans. Syst. Man Cybern. B 38 (5) (2008) 1402–1412, http://dx.doi.org/10.1109/TSMCB.2008.926329.

[38] A. Konak, D.W. Coit, A.E. Smith, Multi-objective optimization using genetic algorithms: A tutorial, Reliab. Eng. Syst. Saf. 91 (9) (2006) 992–1007, http://dx.doi.org/10.1016/j.ress.2005.11.018.

[39] J. Burak, O.J. Mengshoel, A multi-objective genetic algorithm for jacket optimization, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2021, pp. 1549–1556, http://dx.doi.org/10.1145/3449726.3463150.

[40] F.T. Liu, K.M. Ting, Z.-H. Zhou, Isolation forest, in: 2008 Eighth Ieee International Conference on Data Mining, IEEE, 2008, pp. 413–422, http://dx.doi.org/10.1109/ICDM.2008.17.

[41] K.-L. Li, H.-K. Huang, S.-F. Tian, W. Xu, Improving one-class SVM for anomaly detection, in: Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 03EX693), Vol. 5, IEEE, 2003, pp. 3077–3081, http://dx.doi.org/10.1109/ICMLC.2003.1260106.

[42] T. Ahmed, Online anomaly detection using KDE, in: GLOBECOM 2009-2009 IEEE Global Telecommunications Conference, IEEE, 2009, pp. 1–8, http://dx.doi.org/10.1109/GLOCOM.2009.5425504.

[43] B. Golparvar, P. Papadopoulos, A.A. Ezzat, R.-Q. Wang, A surrogate-model-based approach for estimating the first and second-order moments of offshore wind power, Appl. Energy 299 (2021) 117286, http://dx.doi.org/10.1016/j.apenergy.2021.117286.

[44] R. Quevedo-Reina, G.M. Álamo, L.A. Padrón, J.J. Aznárez, Surrogate model based on ANN for the evaluation of the fundamental frequency of offshore wind turbines supported on jackets, Comput. Struct. 274 (2023) 106917, http://dx.doi.org/10.1016/j.apenergy.2021.117286.

[45] S. Zheng, C. Li, Y. Xiao, Efficient optimization design method of jacket structures for offshore wind turbines, Mar. Struct. 89 (2023) 103372, http://dx.doi.org/10.1016/j.marstruc.2023.103372.

[46] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[47] A. Kotelnikov, D. Baranchuk, I. Rubachev, A. Babenko, Tabddpm: Modelling tabular data with diffusion models, in: International Conference on Machine Learning, PMLR, 2023, pp. 17564–17579, http://dx.doi.org/10.5555/3618408.3619133.

[48] L. Prokhorenkova, G. Gusev, A. Vorobev, A.V. Dorogush, A. Gulin, CatBoost: unbiased boosting with categorical features, Adv. Neural Inf. Process. Syst. 31 (2018) http://dx.doi.org/10.48550/arXiv.1706.09516.

[49] Y. Gorishniy, I. Rubachev, V. Khrulkov, A. Babenko, Revisiting deep learning models for tabular data, Adv. Neural Inf. Process. Syst. 34 (2021) 18932–18943, http://dx.doi.org/10.48550/arXiv.2106.11959.