--Manuscript Draft--

Encoding and Synthesizing Part Span Shroud for Aero-Structural Design Optimization using Genetic Algorithm

Mahdi Jalalifar

Department of Mechanical Engineering, Tarbiat Modares University (TMU), Tehran, Iran. E-mail: mjalalimech@gmail.com

Abstract

In recent years, scientific tasks are increasingly taking advantage of Machine Learning (ML) to gain insight and actionable information from data, especially in simulation-driven engineering tasks. Usually, the novelty in the design of aerodynamic shapes relies on the definition of geometric parameters to synthesize new designs, involving computationally intensive simulations to find the feasible design space. Instead, machine learning provides methods to extract information from data that could be translated into knowledge about the underlying concepts of designing, resulting in synthesizing novel shapes with learning representations from existing designs automatically. Moreover, machine learning algorithms can increase domain knowledge and perform tasks related to optimization automatically. In this study, multi-objective optimization of a Part Span Shroud (PSS) of an axial compressor rotor has been performed through a genetic algorithm that isentropic efficiency, Von Mises stress, and total deformation are objective functions. To synthesize a new shroud (PSS), we use a data-driven shape encoding and generating model, called VAEGAN which is a neural network, combining Variational Autoencoder (VAE) with Generative Adversarial Network (GAN). Through this method, designers can encode shape information comprehensively using learned features, without predefined design parameters and then generate novel designs. A combination of interpolating and extrapolating learned feature vectors and sampling from Gaussian noises is used as an initial population in the Genetic Algorithm (GA) to synthesize the PSS that has desired aero-structural properties. By this optimization, Von Mises stress and total deformation decreased by 68 percent and 72 percent respectively, compared to the blade without part span shroud. This happens with the least amount of drop in aerodynamic efficiency which is 0.52 percent.

Keywords: *Machine Learning, Genetic algorithm, Multi-objective optimization, Autoencoding, VAE, VAEGAN, Axial Compressor, Turbomachine.*

1. Introduction

1. 1. Part-Span Shroud (PSS)

Long blades used in modern industrial high-efficiency turbomachines can require the use of a part-span shroud ring to damp blade vibrations originating in natural or forced conditions. However, the existence of a shroud ring at the middle of the blade span can exert detrimental effects on the aerodynamic performance. In earlier designs, adjacent blades were tied together by rods proved to be unsuitable because of stress concentrations at the holes on the blades. Therefore, the modified concepts of PSSs have been introduced. Roberts [1] obtained a correlation between PSS losses and blade geometric data. The data were gathered from 19 tests on research rotors that were transonic fans with part span shrouds. The maximum loss was correlated with the mean inlet Mach number at the shroud location, the geometric parameters of leading- and trailing edge shroud radius normalized by mean passage height and shroud aerodynamic chord, respectively. In [2], the performance of the low-pressure steam turbine with different mid-span shrouds was examined, which is useful to turbine designers for trade-off studies. Wu et al. [3] used an optimization method using Isight to minimize the drag force caused by the PSSs while maximizing the structural life of the blade. Computational Fluid Dynamic (CFD) and structural analysis components are integrated with Isight employing the Simcode component inside Isight. Markus Hafele et al. [4] have numerically optimized the aerodynamic shape of PSSs in the last stage of a low-pressure industrial steam turbine. The results of this study show that the loss depends on the blockage area. They have realized that a change in shape from circular shroud to elliptical reduces additional loss. Jalalifar et al. [5], studied the effect of the shroud's parameters on the aerodynamic performance of a transonic compressor such as drag force and chocks. In the present work, a multi-objective genetic algorithm and autoencoding algorithms have been used to generate the optimum aerodynamic shape of PSS of an axial compressor rotor to enhance its aero-structure performance. In the process of optimization, the principal aims are to minimize stress and tip deformation of the blade while maximizing the isentropic efficiency of the rotor.

1. 2. Shape parameterization

In the design of aerodynamic shapes, geometry parameterization plays a key role [6]. A lot of work has been done in parameterizing complex shapes and reducing geometric dimensions. Some research focuses on analytically expressing the curves. In [7], the design variables of the baseline shape are correlated with shape functions through the coefficients to reach a compact formulation for parameterization. Also, [8] deploys different shape functions to formulate the curve. Apart from formulating the curves by analytical functions, dimension reduction can happen through polynomial and spline. Accordingly, the geometric shape can be described as a linear combination of the basis generated by different orders of polynomials [9], [10]. Another mathematical method for the formulation of curves is the Bezier curve [11]built upon the Bernstein polynomials. Such conventional parameterization techniques or dimension reduction depend on defining parameters manually, so they may work well for a specific geometry but may fail to express complex geometries. The pre-defining the design space and the boundary of the design space can prevent the synthesis of novel samples. In contrast, data-driven methods can automatically learn compact and informative representations from samples. In this study, we utilize a generative model to synthesize novel part span shrouds (called shrouds shortly in this work) without any predefined design parameters. Our deep learning model also automatically learns to parameterize shrouds into latent feature vectors.

2. Methodology

2. 1. Design of Experiments (DOE)

CFD simulations generate accurate flow field predictions which are ultimately used to calculate performance metrics. To generate a dataset including a large number of CFD simulations, an automated workflow is used which builds a Design of Experiments matrix using Latin Hypercube Sampling (LHS), implemented in ANSYS workbench 18.1 [12]. In a Latin Hypercube Sampling, the points are randomly generated in a square grid across the design space, but no two points share the same value, so no point shares a row or a column of the grid with any other point [13].

In the process of optimization in this study, the following criteria have been taken into account:

- Maximize aerodynamic efficiency
- Minimize the stress of the blade
- Minimize blade tip displacement

To calculate the aerodynamic efficiency, Von Mises stress, and total deformation, the one-way FSI (Fluid- Structure Interaction) method is used. Accordingly, aerodynamic loads are calculated in ANSYS CFX to calculate isentropic efficiency and then transferred to the Static Structural application to compute stress and total deformation. We use the generated dataset to train the random forests algorithm to build a supervised model to predict the properties of synthesized shrouds in the optimization process. Random forests are based on an ensemble of decision trees. This algorithm hierarchically splits the data using simple conditional statements, and these decisions are interpretable and fast to evaluate [14].

2. 2. Shape Representations

I. Dimension Reduction

One of the difficulties in design optimization based on CFD simulations is related to the curse of dimensionality [15]. Therefore, before applying any design optimization methods, the design domain should be defined by dimension-reduction. Usually, designers select design parameters manually using traditional parameterization techniques introduced in the introduction, which impose restrictions on the generalization and novelty of the design. To address this problem, self-supervised deep learning methods like Variational Autoencoder (VAE) [16] and Generative Adversarial Network (GAN) can be implemented on dimension reduction, which synthesize shapes with novelty. In the current work, we introduce VAEGAN [17] exploiting both VAE and GAN to learn latent features of samples and then generate novel shrouds without any predefined design parameter. The VAE model is trained to encode a high-dimensional existing sample's shape into a low-dimensional representation, and then reconstructs the shape with little error through a well-trained decoder network. By minimizing the difference between the reconstructed sample and the original one, VAE automatically learns features without any labels. A GAN uses a discriminator to tell us whether the samples are real or not. The cross-entropy between real and fake calculated using the discriminator's output is fed back to the generator as a signal to learn to produce realistic samples. These self-supervised learning methods help the parameterization of various geometries using the learned informative features without restrictions from human labels. So far, autoencoders have been mainly used in image processing [18], and their generative capabilities are regularly seen in generating complex information such as realistically looking human faces (deep fakes). Recent state-of-the-art modeling in the aerospace field involves the application of surrogate models [19] to perform design optimization. The presented application of autoencoders can be seen as a surrogate, although the generating approach differs from typical polynomial, kriging, or gaussian process-based models [20].

II. Autoencoders' Structure and Loss Functions

As mentioned above, we introduced VAEGAN [17] [21] as a method to synthesize realistic novel shrouds, consisting of VAE, and GAN models. Three models are generative probabilistic autoencoders, meaning that they can generate new instances determined by chance [22]. Here, we explain the structure of each model along with their cost functions that lead them to produce desired samples.

Variational Autoencoder (VAE)

VAE's structure is simply an encoder followed by a decoder. The former encodes a high-dimensional sample x, into a mean coding μ and a standard deviation σ, that is low-dimension, and then actual coding (low-dimensional latent representation, z) is represented randomly from a Gaussian distribution with mean μ and standard deviation σ. After that, the decoder takes the latent vector, z (coding) as an input, and decodes it from the representation domain (latent space or coding space) to the original data domain, $\tilde{x}$ [22]. The encoder and decoder are given as:

$$\boldsymbol{z \sim Enc(x) = q(z|x),} \tag{1}$$
$$\boldsymbol{\tilde{x} \sim Dec(z) = p(\tilde{x}\,|z),} \tag{2}$$

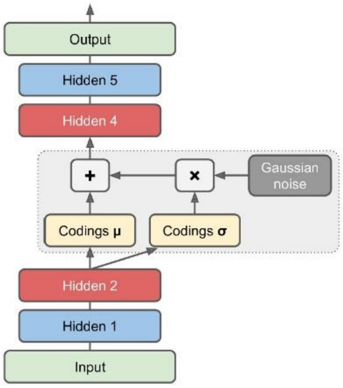


Figure 1. Variational autoencoder [22]

The cost function for VAE is composed of two parts. The first is reconstruction loss which pushes the model to reproduce itself, which is the mean squared error (MSE) here. The second is the latent loss (known prior loss) that pushes the model to have latent codings (feature vector) sampled from a simple Gaussian distribution, which is a KL divergence between Gaussian distribution and the actual distribution of features. Variational autoencoder's loss function to minimize is given by:

$$\boldsymbol{L_{VAE}} = \boldsymbol{L_{\text{recon}}} + \boldsymbol{L_{prior}}\,, \tag{3}$$

$$\boldsymbol{L_{\text{recon}}} = \|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_2^2\,, \tag{4}$$

$$\boldsymbol{L_{latent}} = \boldsymbol{D_{KL}(q(z|x)\|p(z))} = -\frac{1}{2}\sum_{i=1}^{n} \boldsymbol{1 + \gamma_i - \exp(\gamma_i) - \mu_i^2}, \tag{5}$$

In the equation of reconstruction loss, Eq.4, x, and $\tilde{x}$ are the original sample and the reproduced one respectively, and in the latent loss equation, Eq.5, n is the codings' dimensionality, μ and σ are the mean, and standard deviation of the i component of the codings, and $\gamma = \log \sigma^2$, [22].

Generative Adversarial Network (GAN)

GAN [18] contains a generator (G) and a discriminator (D), competing with each other in a self-supervised manner. D learns to determine synthesized fake samples from real ones, while G tries to generate realistic samples to fool D. The generator takes random noise $\hat{z} \sim p(\hat{z})$ as input to generate fake $\hat{x}$. The discriminator takes both x and $\hat{x}$ to predict whether the input is from the real dataset or generated by G. The cross-entropy loss function for the min-max game is given as: [21]

$$\textbf{min max } \boldsymbol{L_{\text{GAN}}} = \log \boldsymbol{D(x)} + \log(1 - \boldsymbol{D(G(\hat{z}))}), \tag{6}$$

VAEGAN

The GAN generates samples purely from random noise. In other words, the generator never actually sees any real images, yet it gradually learns to produce convincing fake images. All it gets is the gradients flowing back through the discriminator. This makes it hard to obtain an explicit mapping from the data domain to the feature domain [22]. Also, the VAE suffers from poor performance in generating novel samples that have not been seen before [8] [17]. To address these challenges of these two models, the VAEGAN has been introduced, which has an encoder-decoder structure from VAE and a discriminator from GAN, as shown in Figure 2.

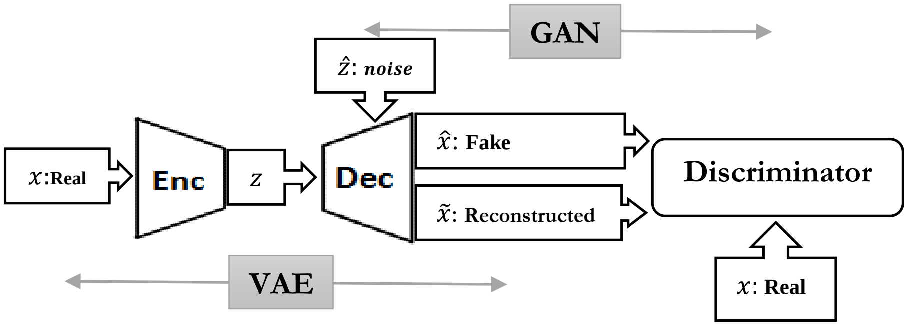


Figure 2. VAEGAN model combining VAE and GAN models

In the process of VAEGAN training, we reconstruct a sample, $\tilde{x}$, which is a copy of a real given sample, x, and also generate a fake sample, $\hat{x}$, directly from noise, $\hat{z}$. Then we determine both $\tilde{x}$ and $\hat{x}$ as fake (zero) for the discriminator, and only x is defined as the real sample (one). GAN discriminator loss function, L_{GAN} is the binary cross entropy:

$$\boldsymbol{L_{\mathrm{GAN}} = \log D(x) + \log(1 - D(Dec(z))) + \log(1 - D(Dec(Enc(x))))}, \tag{7}$$

Besides the element-wise reconstruction loss function (mean squared error) used for VAE, we introduce another loss function, L_{layer} , that is a feature-wise error to better capture the complex data distribution [17] and to stabilize the training process [21]. L_{layer} , as given in Eq. 7, is expressed in the discriminator, and measures the l_1 distance between the mean values of the neurons in one particular layer of D when fake samples/reconstructed samples are fed and the values when real samples are fed.

$$\boldsymbol{L_{layer\_fake} = \|D_l(x) - D_l(Dec(\hat{z}))\|_1}, \tag{8}$$

$$\boldsymbol{L_{layer\_reconstructed} = \|D_l(x) - D_l(Dec(Enc(x)))\|_1}, \tag{9}$$

The complete loss function for VAEGAN is given by:

$$\boldsymbol{L = L_{latent} + L_{\mathrm{recon}} + L_{\mathrm{GAN}} + L_{layer\_fake} + L_{layer\_reconstructed}}, \tag{10}$$

The three components, encoder, decoder, and discriminator are trained jointly.

2. 3. Optimization Procedures

I. Genetic Algorithm (GA)

In recent years, instead of using trial-and-error approaches for turbomachinery design, search-based optimization techniques combined with CFD are widely used to reduce experimental and computational expenses. To evolve shrouds for reaching a suitable one possessing desired properties, a genetic algorithm (GA) [23] is deployed. The GA is a population-based optimization technique based on the Darwinian theory of survival of the fittest [24]. According to the review article [25] which compares gradient-based and stochastic optimization algorithms for aerodynamics, the process of GA optimization is summarized as:

1) **Generation** of individuals to form the initial population.
2) **Evaluation** of the fitness of each individual in the population.
3) **Selection** of individuals to take part in genetic operations.
4) Creating a **new population** by applying genetic operations such as crossover and mutation.
5) **Iterate** over steps 2–4 until some convergence criterion is met.

Accordingly, after generating the initial population of individuals as the candidate solutions, the first iteration of the algorithm is started. All candidates are evaluated using the objective function. Then parents are selected based on their fitness in tournament selection, which are used as the basis for generating the next generation of candidate vectors. Crossover is used to create children from parents' genetic information and inherit their traits, also mutation is used to introduce new genetic information. Mutation allows the offspring to evolve through un-inherited characteristics, which promotes diversity that prevents the GA from converging to local optima.

II. Synthesized Shape Optimization

Generating novel shrouds is not the end, and we aim to design one that has suitable aero-structural properties. To this end, learned feature vectors evolve through the GA to reach the shroud with desired properties. As mentioned above, this gradient-free optimization technique is inspired by natural selection and is intended to force individuals to gradually evolve. Initially, our individuals are the learned feature vectors (z) that can be interpolated, extrapolated, sampled, or a combination of all of them. We have a population including a certain number of $z_{i,j}$ that means feature vector for j^{th} individual in the i^{th} generation. This population is updated in each generation (every iteration) by replacing selected children that are generated using single-point crossover and mutation. By combining the GA and a well-train random forests algorithm that predicts aero-structural properties, we can have low-cost and high-speed calculations. The fitness score, $f_{i,j}$, is the difference between the target (eff_t) and current efficiency ($eff_{i,j}$) and normalized by the target, as shown in:

$$\boldsymbol{f_{i,j}} = \frac{\boldsymbol{eff_{i,j} - eff_t}}{\boldsymbol{eff_t}}, \tag{11}$$

where, $eff_{i,j}$ represents efficiency of $s_{i,j}$. The fitness score is supposed to approach zero as individuals evolve with each generation. The decoded individual that has a lower fitness score wins the tournament and becomes one of the parents. Besides, we have two other objects that are minimum stress and minimum total deformation, which should be met to select the individual as the best candidate in each iteration. After choosing the appropriate parents, children are generated by implementing single-point crossover. Accordingly, all elements after a random point are swapped between the two parents. Mutation in the natural selection process is also imitated with additive Gaussian noises.

3. Experiment

3. 1. Geometry Modeling Approach

Part span shroud was originally designed as a simple cylinder of material between two blades to decrease Von Mises stress on the blade which increases the life cycle. The negative impact of PSS on the flow field should be minimized, therefore, the geometry must be aerodynamic. In [1], the typical geometries of part-span shrouds were introduced (Figure 3a). In the present work, the properties of the optimum shroud as a final result will be compared to a simple cylindrical rod and the type "A" which is based on two ellipses joined by straight tangent lines. Figure 3b shows the geometry of the compressor blade which is used in this study. The characteristic of this rotor is presented in Table 1. The geometry was built with the help of the Design Modeler (DM) module in ANSYS Workbench.

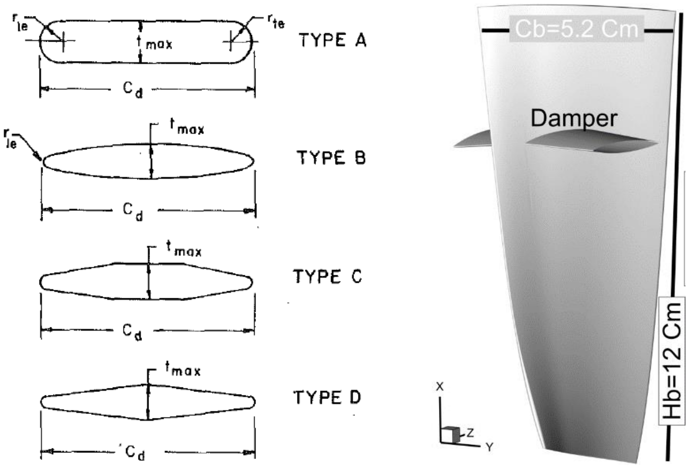


Figure 3. (a) Typical geometries of PSS [1], (b) Blade geometry in the current work

Table 1. The characteristic of the rotor

Character	Number of blades	Average height	Chamber line	Aspect ratio	Rotor speed	Tip speed	mass flow
Value	25	12 Cm	2.5 Cm	2.4	1741 (rad/s)	348.2 m/s	24.1 kg/s

3. 2. CFD Approach

The commercial code ANSYS CFX 18.1 is used for flow analysis. To consider the effects of inlet and outlet flow distortions on the general flow pattern, the domain should be extended. In this study, the upstream and downstream fluid domains were set to about four and eight times the mean chord, respectively [26]. The $k-\omega\, SST$ turbulence model was utilized. The inlet boundary conditions were prescribed to the total pressure in 101325 Pascal, and the direction of the inlet velocity was set to "normal to boundary", assuming no inlet swirl. At the outlet boundary, the static pressure was specified in 104000 Pascal which was less than the stall pressure investigated in the previous work [27]. To reduce total mesh number and computational time, one blade channel, along with applying periodic conditions for the lateral surface of the domain were utilized. The surfaces of the hub, shroud, and blade are treated as the walls with the no-slip condition. The Mesh study in Figure 4 shows that around 800000 elements can perfectly satisfy mesh independency conditions.

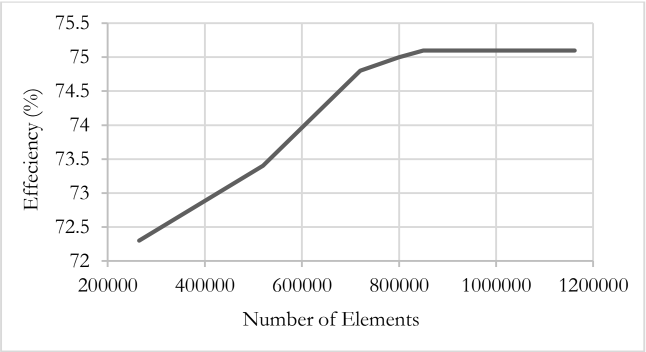


Figure 4. Mesh study

3. 3. Structural Approach

The solid geometry of the PSS body was constructed using the Design Modeler module which was integrated with another ANSYS workbench component, called Static Structural analysis. In the assumed structure, the tip of the part span shroud was set to freely move in radial and axial directions, meaning that the blade can slightly alter in these directions. This assumption is necessary as otherwise, stress would increase dramatically at the edge of PSS. PSS is fixed at the interface of two adjacent part span shrouds which is not considered due to one blade assumption. In the direction of axis normal, the blade cannot vibrate.

3. 4. Data Pre-processing

The database provided by DOE is represented by numbers of points with x, and y coordinates used to train the generative model. We first scale the x coordinates of all samples to [0, 1] to have homogenous inputs fed into the neural network directly, then are interpolated by splines. Therefore, all the interpolated samples share the same x coordinates. As a result, only the y coordinates of each sample are fed into the model. Finally, all y coordinates are scaled to [-1, 1] (Figure 5).

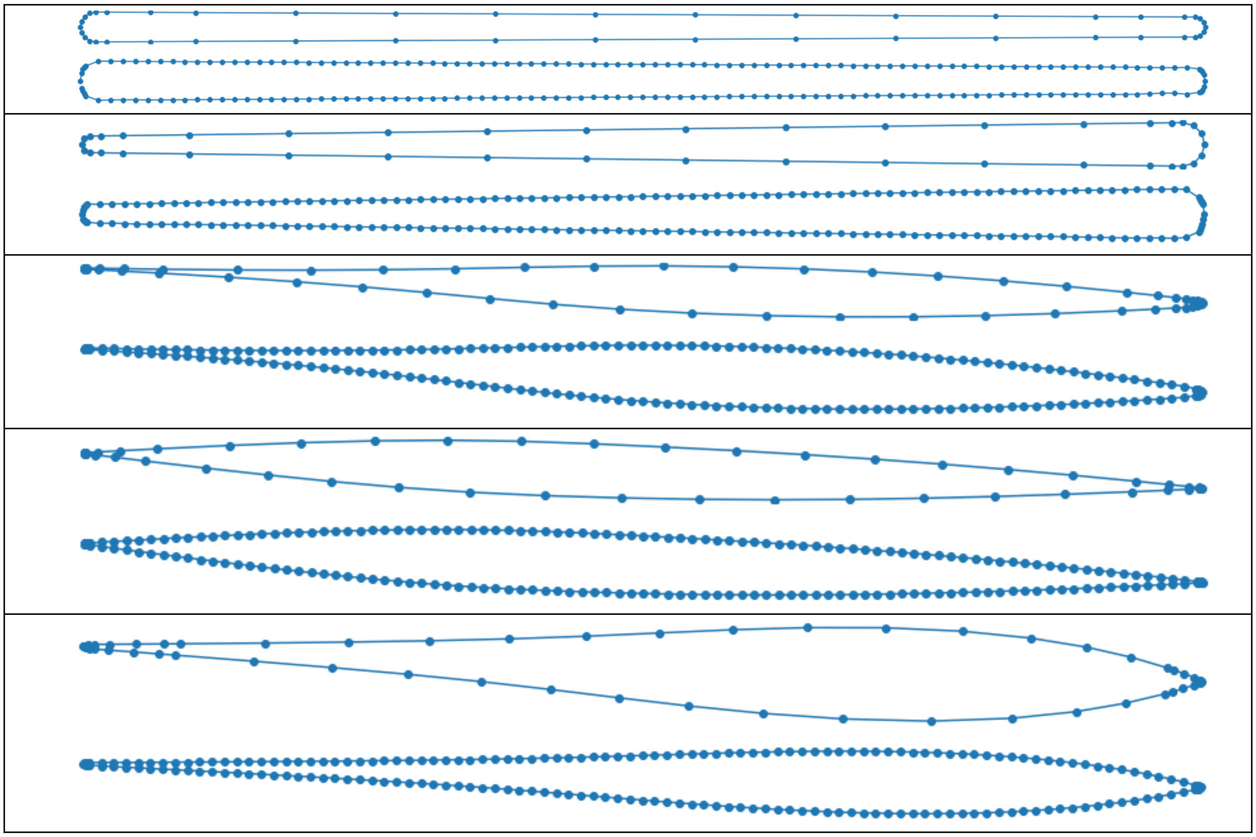


Figure 5. In each section, the first row shows the origin PSS coordinates from the database computed by DOE, the second row shows the corresponding processed PSS

3. 5. Preparing the model

An encoder, a decoder, and a discriminator built on 3 multi-layer perceptron (MLP) are combined to model the VAEGAN. Inputs which are 200-dimensional shroud coordinates are encoded into a 10-dimensional feature vector transferring to the decoder to map to the shroud. Then, the result sends to the discriminator as a classifier to judge whether the production is a real shroud from the database, a copy reconstructed from the decoder or synthesized from random noises. LeakyReLU [28] is implemented as the activation function in each layer except in the last layer of the discriminator that has a Sigmoid activation function, and output layer in the decoder the hyperbolic tangent (Tanh) function works as the activation function to scale all outputs into [-1, 1]. Our VAEGAN model needs to train on the database for 5000 epochs with an initial learning rate of 0.0005 using the Adam optimizer [29] to learn how to extract the informative representations and generate a realistic shroud.

3. 6. Comparison between, PCA, VAE, GAN, and VAEGAN

Here, there is a comparison between PCA, VAE, and VAEGAN. Principle Component Analysis (PCA) conducts a linear transformation from the pre-processed sample point coordinates into prioritized latent variables. The latent feature dimension is also set to 10 for all three models. Also, we can compare the accuracy of the reconstruction power of VAE and VAEGAN. As Figure 6 shows, synthesized shrouds using PCA are useless, unlike two other methods that have acceptable quality. However, the last comparison between VAE and VAEGAN, which reproduced one specific shroud, demonstrates the high accuracy of the VAEGAN model. Finally, we can implement smoothing filters to smoothen the boundary of reconstructed and fake shrouds. In our case, the second-order polynomial is used in the Savitzky-Golay filter [30], which is a moving polynomial fitting, and the length of the moving window is set to be 7. There was no big change or improvement in our generated samples generated through VAE and VAEGAN, But the GAN model that had bad quality before using the filter can be smoothed after using the filter. Although the filter could smoothen the boundary to some extent, it was not good enough to use the GAN model in the optimization process.

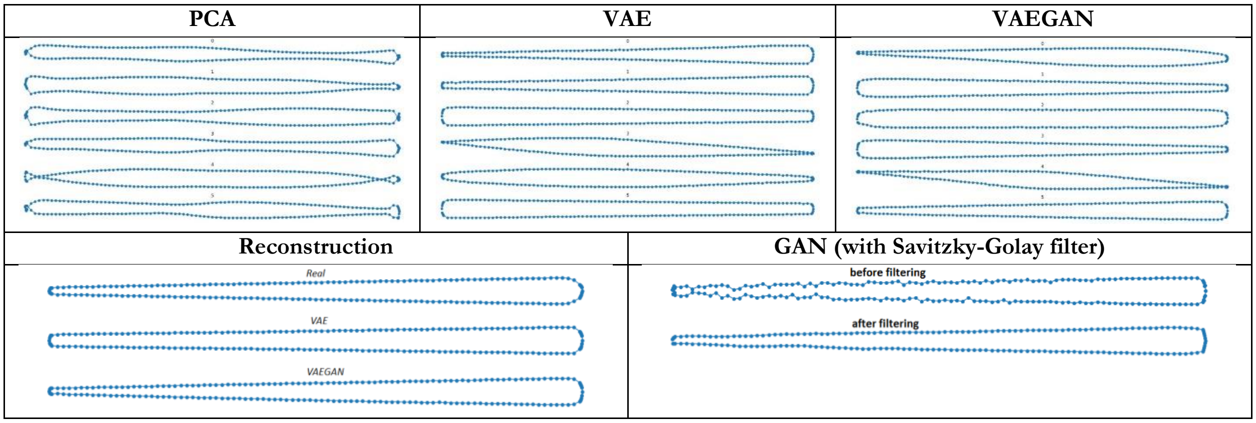
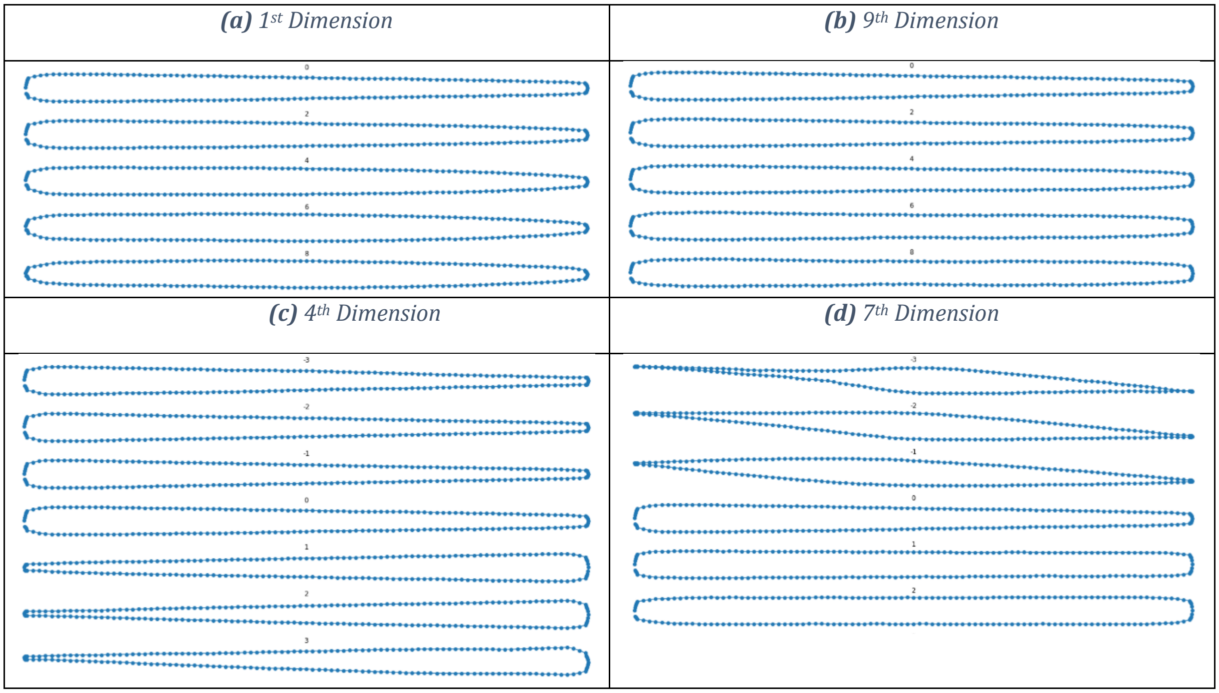


Figure 6. Comparison: PCA, VAE, GAN, and VAEGAN

3. 7. What is encoded in the Feature Domain?

The ability of encoding high-dimensional geometries in low-dimensional feature vectors provides us manipulating geometries by changing one specific element. To achieve this, all the elements are set to zero except for one specific dimension. That particular element is changed gradually, and the designed feature vectors are mapped to the shroud coordinate domain by the decoder [21]. Figure 7 shows the basic shape of a sample with feature vector including zeros, and changes in the 1st, 4th, 7th, and 9th dimensions respectively. As the 1st dimension is increasing from 0 to 8, the right side changes from line to curve, meaning that the thickness of the shape from the middle side to the right side grows. Similarly, this happens when the 9th dimension is changing from 0 to 8. The 4th and 7th dimensions are sensitive to the sign of elements, meaning that changes in shapes get inverse when the sign of elements changes from negative to positive. In detail, by decreasing the 4th dimension from 0 to -3, we can see that the right side of the generated shape becomes thinner, while the left side becomes thinner when increasing from 0 to 3. Changing both sides of the sample can happen simultaneously if we manually make alterations to the 7th dimension. If this dimension decrease from 0 to -3, both corners become thinner, in contrast, they become wide if the element's value climbs from 0 to +3.

In comparison to the features learned by the VAEGAN model, Figure 8 shows the VAE-synthesized shrouds when changing only one feature dimension. Interestingly, we can see that what happened in VAEGAN is seen in VAE, but there are some differences in details.



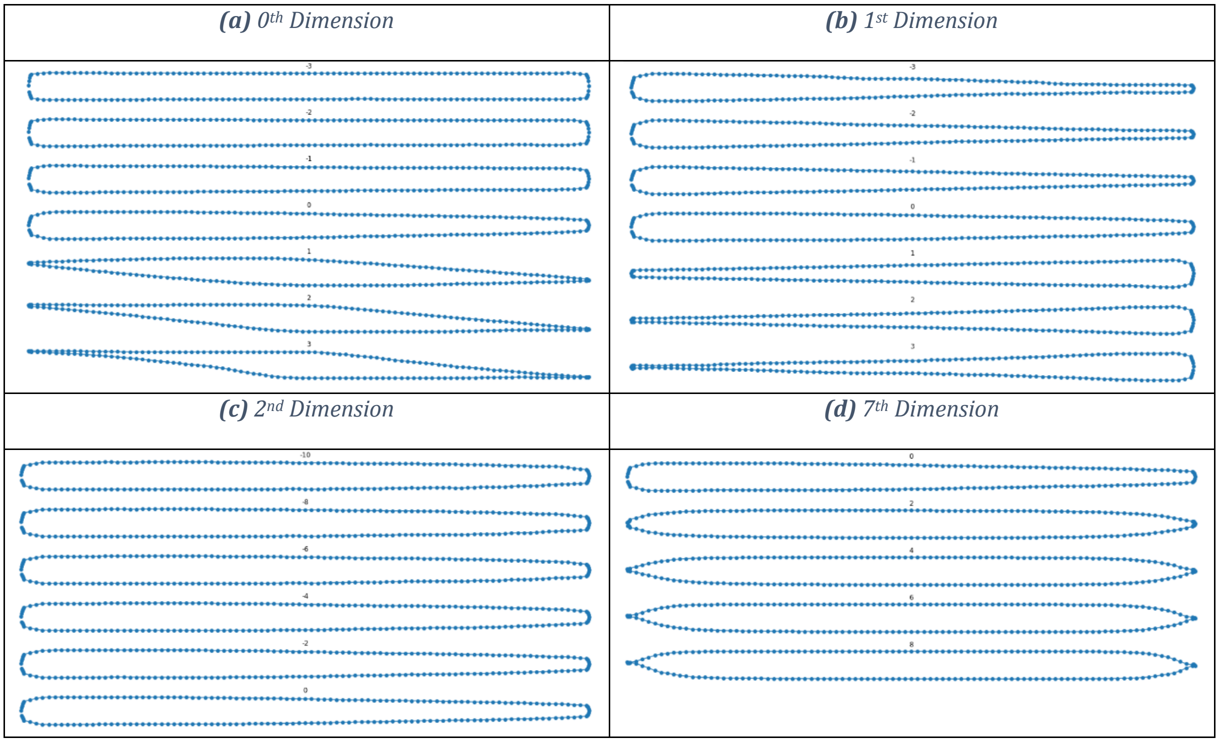
(a) *1st Dimension*

(b) *9th Dimension*

(c) *4th Dimension*

(d) *7th Dimension*

Figure 7: VAEGAN-generated shrouds by gradually changing only one dimension of the feature domain: (a) changes the 1st dimension, (b) changes the 9th dimension, (c) changes the 4th dimension, (d) changes the 7th dimension



***(a)** 0th Dimension*

***(b)** 1st Dimension*

***(c)** 2nd Dimension*

***(d)** 7th Dimension*

Figure 8. VAE-generated shrouds by gradually changing only one dimension of the feature domain: (a) changes the 0th dimension, (b) changes the 1st dimension, (c) changes the 2nd dimension, (d) changes the 7th dimension

3. 8. Synthesis methods: Interpolation, Extrapolation, and Sampling

After training the VAEGAN model on the database, the model learns to encode shape into latent informative features which can be utilized for synthesizing novel shrouds that are different from samples in the training dataset. To achieve this, we can use three synthesis methods: interpolation, extrapolation, and sampling. In interpolation or extrapolation, we use the well-trained encoder to map two real shrouds from the database to latent feature vectors, z_1 and z_2, and then a new feature vector $\bar{z}$ is calculated by combining z_1 and z_2, as given in Eq. 12:

$$\bar{z} = \vartheta * z_1 + (1 - \vartheta) * z_2, \quad (12)$$

$$0 < \nu < 1,\ \ \bar{z}\text{: interpolation, else } \bar{z}\text{: extrapolation}$$

The interpolated/extrapolated feature vector, $\bar{z}$, is then fed into the decoder to synthesize a new shroud. In Figure 9 and Figure 10, interpolated shrouds and extrapolated ones are shown, respectively. In sampling, unlike interpolation or extrapolation, which relies on feature vectors from existing samples, shrouds generate directly from random noise using the decoder. A random vector $\hat{z}$ is sampled from a Gaussian distribution N(0. **I**) and then mapped to a shroud via the decoder (Figure 6). Using the sampling method, we can synthesize a large number of novel samples that we want, because latent vectors are not constrained by features extracted from shrouds in the database.

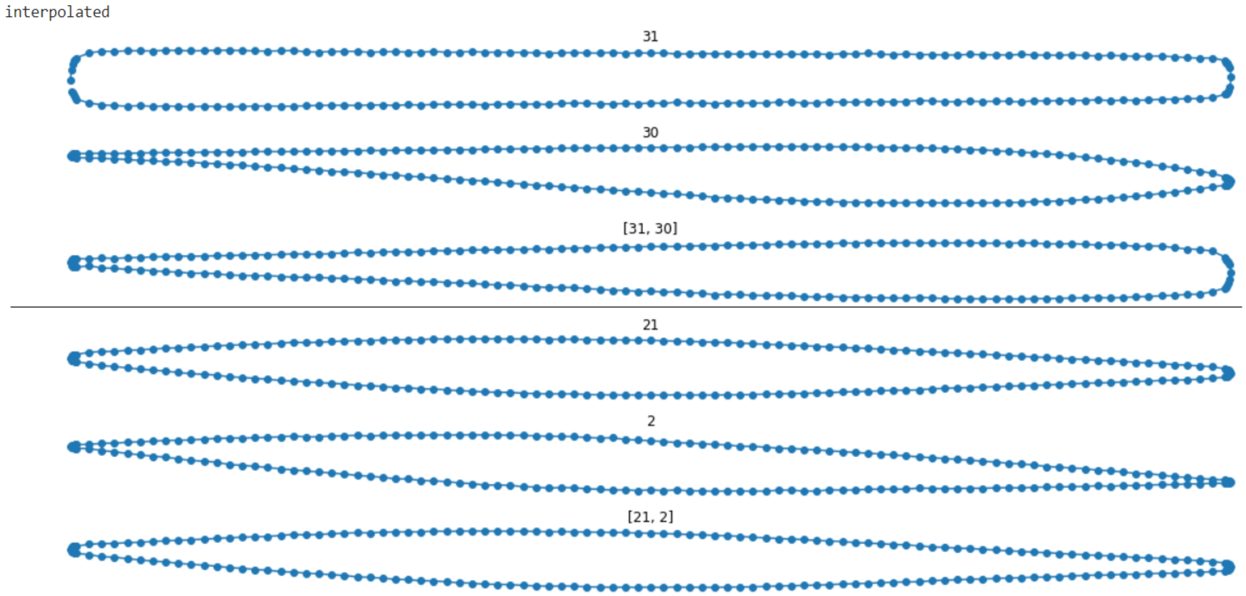


Figure 9. VAEGAN-generated shrouds by interpolation from different feature vectors

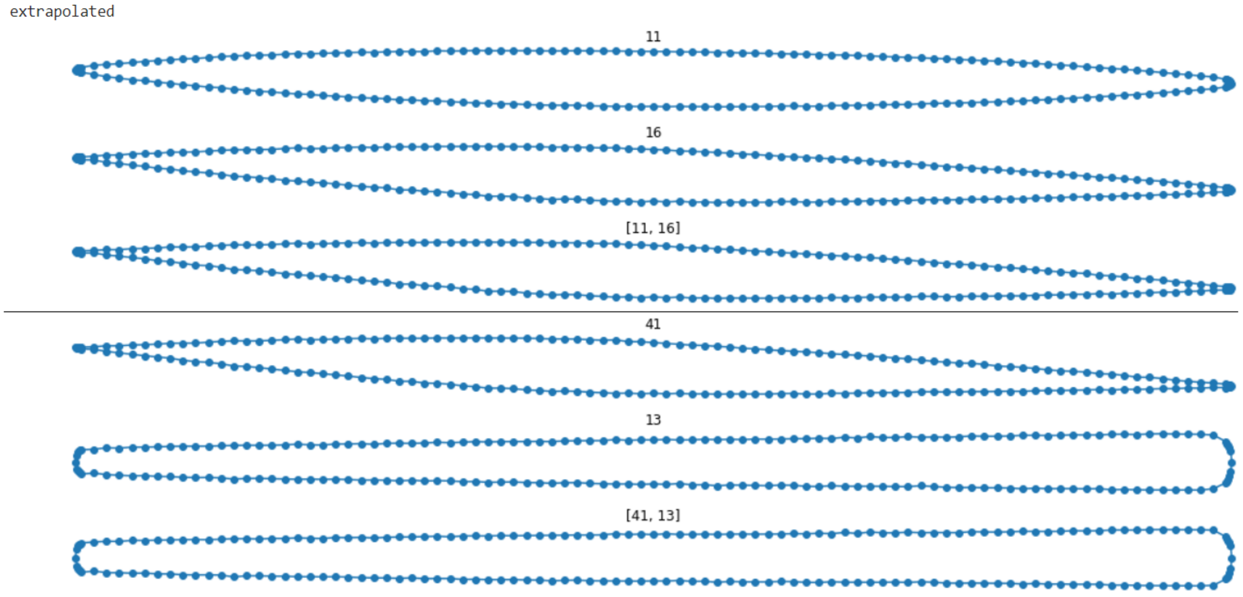
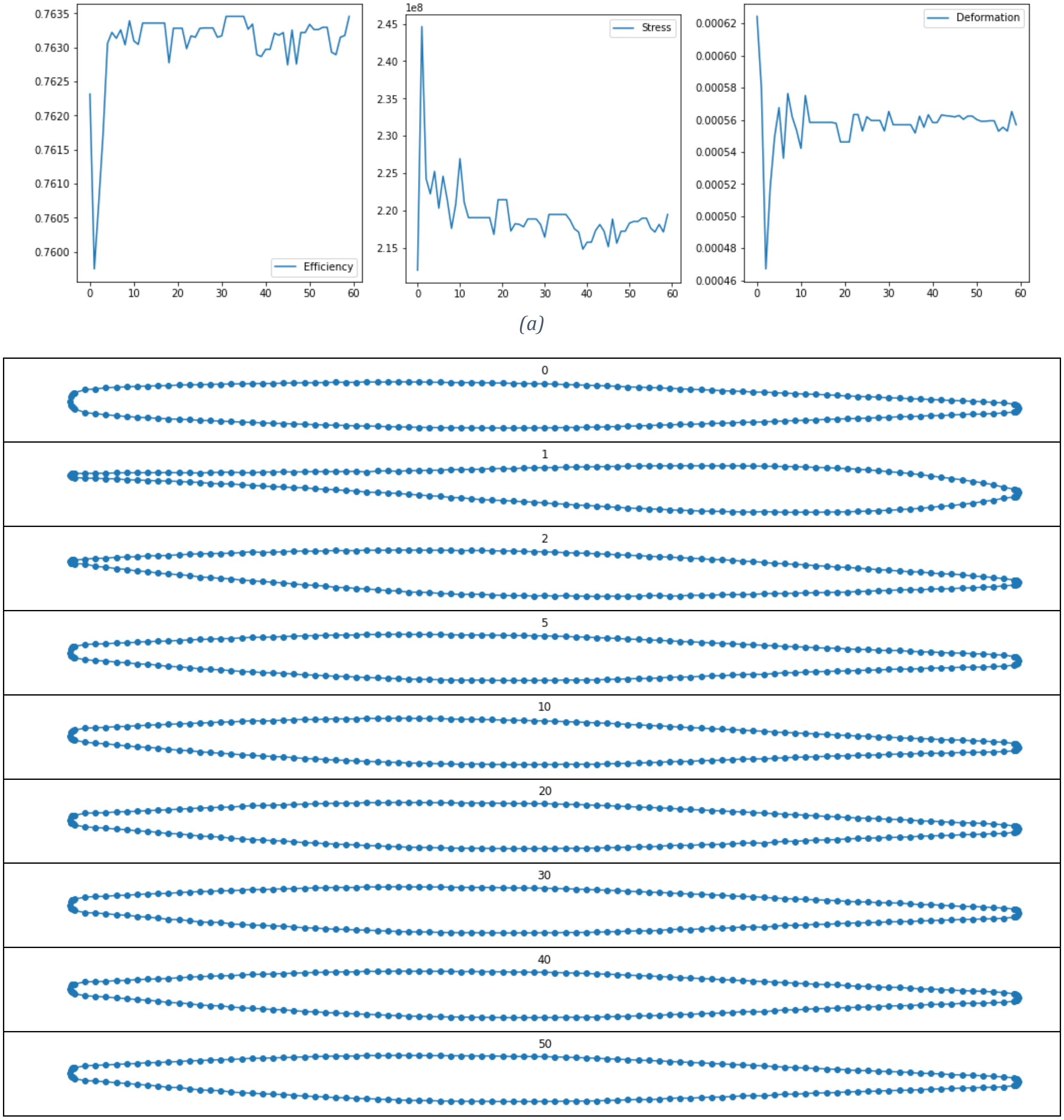


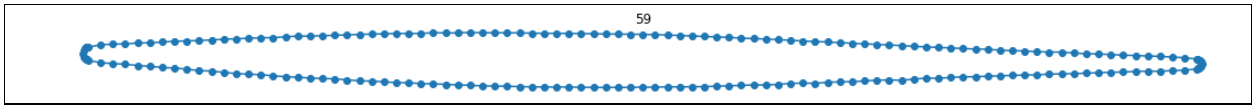
Figure 10. VAEGAN-generated shrouds by extrapolation from different feature vectors

3. 9. Genetic Algorithm and Results

In addition to the target efficiency which is 0.77, two criteria should be taken into consideration: minimizing the stress of the blade and minimizing blade tip displacement. In our case, the total number of generations, N, is set to be 60, and the length of populations (the number of individuals) on each generation, M, is 120 which is a combination of interpolated, extrapolated, and random noise feature vectors. Figure 11a shows how the efficiency, stress, and total deformation change with generation. As Figure 11b illustrates, the shroud's shape gradually evolves to the target. Additionally, we can find the results of two other methods, VAE and PCA in Figure 12 and Figure 13. Finally, in Table 2, we compare the performance of shroud optimization using different techniques, PCA, VAE, and VAEGAN with the genetic algorithm with the same objective function and settings. Comparing these figures and table demonstrates that the VAEGAN model is the best model here to achieve the desired aero-structure properties. Whereas PCA and VAE fail to synthesize the desired shroud within the same number of generations and population size. This is because our VAEGAN-based model generates a wider variety of samples that serves as potential candidates for design optimization. The VAEGAN-synthesized samples reach an efficiency, stress, and deformation of 0.763, 2.2e+08, and 5.6e-04, respectively.

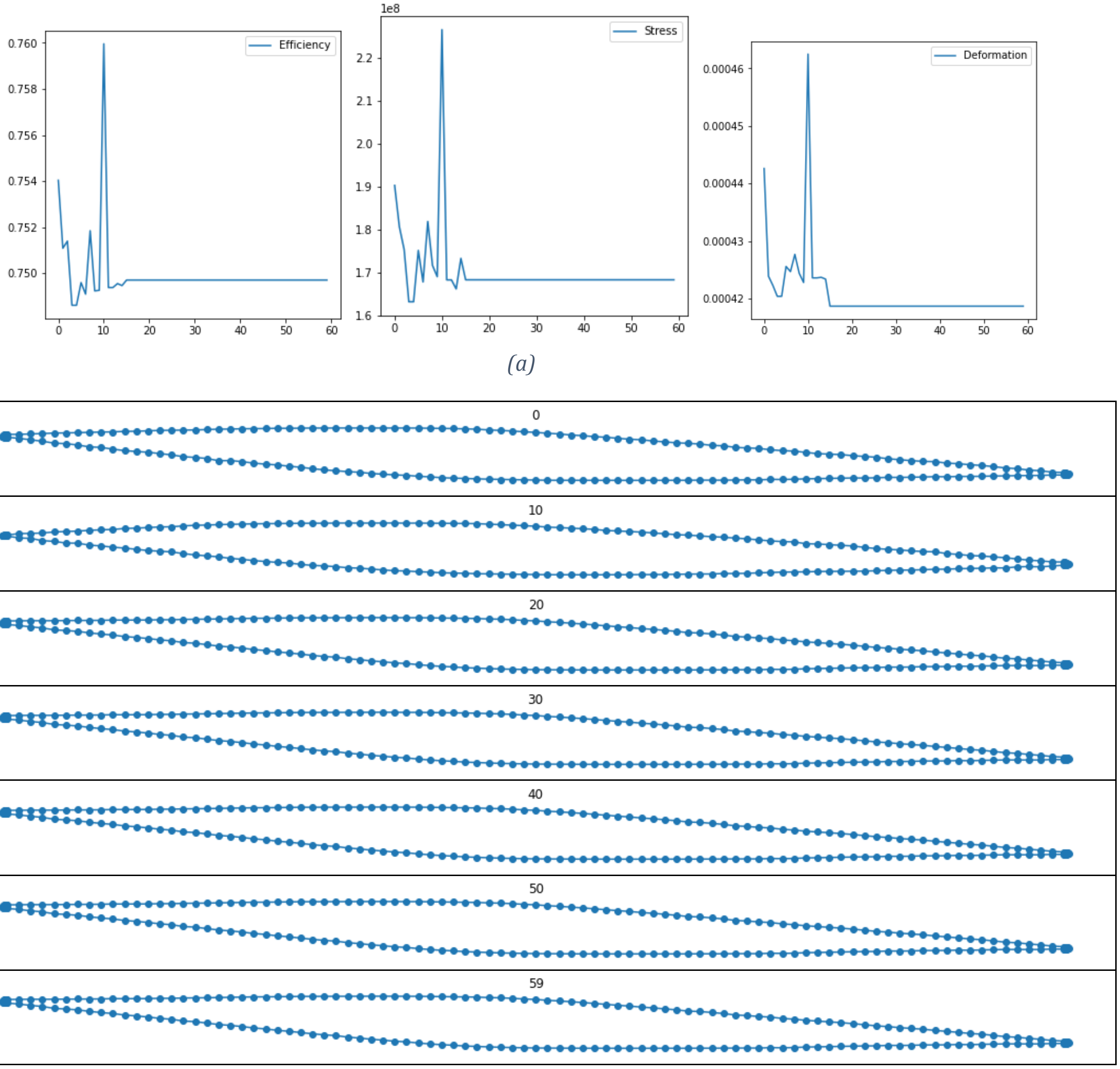


(a)



(b)

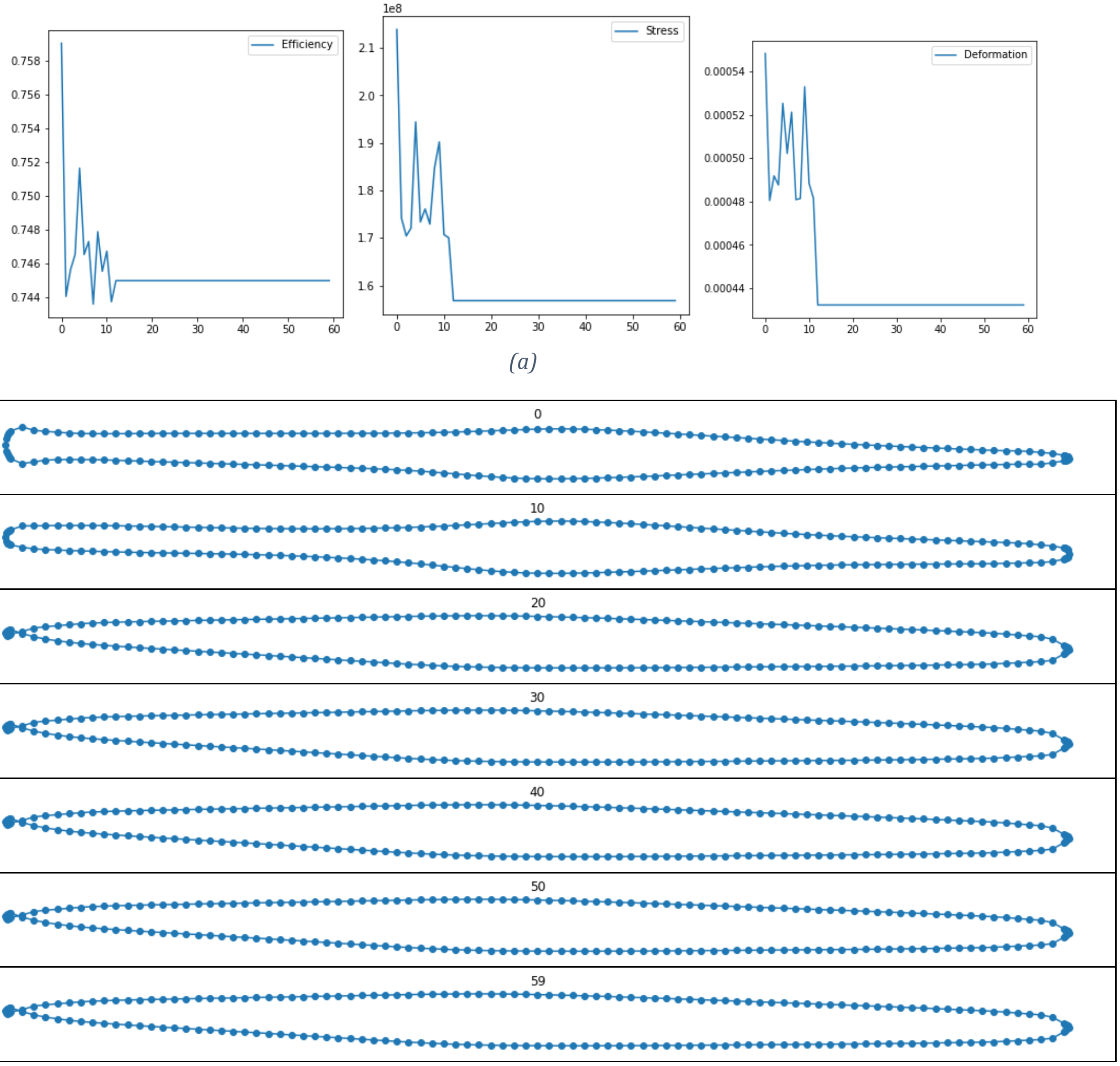
Figure 11. Shape optimization via a genetic algorithm using the VAEGAN model: (a) trend of evolution in efficiency, stress, and deformation of synthesized shrouds over generations; (b) optimized shroud geometry for different generations



(a)

(b)

Figure 12. Shape optimization via a genetic algorithm using the VAE model: (a) trend of evolution in efficiency, stress, and deformation of synthesized shrouds over generations; (b) optimized shroud geometry for different generations



(a)

(b)

Figure 13. Shape optimization via a genetic algorithm using PCA: (a) trend of evolution in efficiency, stress, and deformation of generated shrouds over generations; (b) optimized shroud geometry for different generations

Table 2. Shroud design optimization results with different techniques

	Efficiency		**Stress**		**Deformation**	
Methods	mean	std	mean	std	mean	std
PCA	0.7454	0.0021	1.6e+08	1.1e+07	4.5e-04	3e-05
VAE	0.75	0.0015	1.7e+08	8.4e+07	4.2e-04	6.5e-06
VAEGAN	0.7631	0.0006	2.2e+08	4.1e+07	5.6e-04	1.7e-05

As mentioned above, the initial PSSs were simple rods, and also the typical shape of PSSs is the type A introduced in [1]. In Table 3, there are properties of these two models of shrouds and the optimized one along with the values for blade without PSS, which allows us to compare the results and find out our shroud has the best results. This shroud generated through GA and VAEGAN methods decreases stress and total deformation of the blade by 68 percent and 72 percent, respectively, while just 0.5

percent decreases aerodynamic efficiency. In the other words, if we use this optimum shroud, we can increase efficiency by almost 0.8% and 0.9%, diminish stress by around 50% and 2%, and decrease deformation by 42% and 64% in comparison with the typical shrouds (type A) and rod, respectively.

Table 3. Comparison between different types of PSS and the optimized one

Type	Efficiency	Stress	Deformation
Without PSS	0.767	6.82E+08	2.00E-03
GA + VAEGAN	**0.763**	**2.20E+08**	**5.60E-04**
Typical (type A in [1])	0.757	4.33E+08	9.71E-04
Rod	0.756	2.25E+08	1.58E-03

Such experiments prove that a simple optimization technique like GA and the well-trained VAEGAN model can synthesize samples with desired properties, which can guide the designing of effective and efficient aerodynamic products.

4. Conclusion

In this work, a data-driven method is proposed to achieve three goals: (1) automatically featuring geometries from the database without manually designed parameters, (2) synthesizing novel shrouds by interpolating and extrapolating the encoded features, as well as generating from random noise, and (3) optimizing the features to synthesize samples with desired aero-structural properties. Our model is built upon VAEGAN, which combines the encoder-decoder architecture from VAE and the discriminator from GAN. With the encoder-decoder structure, our model learns how to map from shroud coordinates to the latent feature domain as well as from the feature vectors to the shrouds, while with the discriminator, the model can judge the sample and help the decoder to synthesize realistic samples. With the help of GA, the synthesized shrouds can evolve to have desired properties. Through this method of dimension reduction, we also study the role of each encoded dimension of the feature domain, which represents geometric information like height, radius, and symmetry. Moreover, novel shrouds are synthesized by interpolating and extrapolating learned features from different shrouds, which inherit features from existing shrouds. Also, shrouds can be generated from Gaussian noise, which introduces more novelty. Finally, the synthesized shrouds can be optimized via GA to possess competitive or even better aero-structural properties in comparison to existing ones. The results of this article indicated that autoencoders can be used to perform design optimization with higher accuracy as well as reduce the dimensionality and computational complexity of the optimization problem.

Data availability statement

The related code will be available at https://github.com/mahdi6788/Genetic-Algorithm-VAEGAN.git

5. References

[1] W. B. Roberts, “A Design Point Correlation for Losses due to Part-Span Dampers on Transonic Rotors.” ASME. J. Eng. Power: 101(3), p. 415-421, 1979. https://doi.org/10.1115/1.3446594

[2] Liu J, Mistry H, Santhanakrishnan M, Stein A, Dey S, Slepski J., "Aerodynamic Performance Assessment Of Part-Span Connector Of Last Stage Bucket Of Low Pressure Steam Turbine," Proceedings of the ASME 2011 Power Conference, vol. 1, no. 55265, pp. 545-550, 2011. https://doi.org/10.1115/POWER2011-55265

[3] Moine Y. Wu, J, Marra J. J, Ting Wu Y, Funk C. K, Hsu P. F, Zhou R, Subramanian C. S, Campbell C. X, "Design Optimization of Turbomachinery Components with Independent FEA and CFD Tools in an Optimization Software Environment - a Mid-Span Shroud Ring Study Case," Energy Systems Analysis, vol. 4, no. IMECE2011-62083, p. 8, 2012. https://doi.org/10.1115/IMECE2011-62083

[4] Markus Häfele, Christoph Traxinger, Marius Grübel, Markus Schatz, Damian M Vogt, Roman Drozdowski, "Numerical and experimental study on aerodynamic optimization of part-span connectors in the last stage of a low-pressure industrial steam turbine," Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy, vol. 229, no. 5, pp. 465 - 476, 2015. https://doi.org/10.1177/0957650915591906

[5] Jalalifar Mahdi, Fallah Saleh and Ghadiri Dehkordi Behzad., "Effect of Geometric Parameters of Part Span Damper on Aerodynamic Operation in a Transonic Turbomachine.," Journal of the Chinese Society of Mechanical Engineers , vol. 41, no. 4, pp. 479-491, 2020.

[6] Samareh, J. A., "Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization," AIAA journal, vol. 39, no. 5, p. 877-884, 2001. https://doi.org/10.2514/2.1391

[7] Hicks, R. M., and Henne, P. A., "Wing design by numerical optimization," Journal of Aircraft, vol. 15, no. 7, p. 407-412, 1978. https://doi.org/10.2514/3.58379

[8] HAGER, J., Eyi, S., and Lee, K., "A multi-point optimization for transonic airfoil design," 4th Symposium on Multidisciplinary Analysis and Optimization, p. 4681, 1992. https://doi.org/10.2514/6.1992-4681

[9] Elliott, J., and Peraire, J., "Practical three-dimensional aerodynamic design and optimization using unstructured meshes," AIAA Journal, vol. 35, no. 9, p. 1479-1485, 1997. https://doi.org/10.2514/2.271

[10] TAYLOR, A., III, HOU, G., and KORIVI, V., "Sensitivity analysis, approximate analysis, and design optimization for internal and external viscous flows," Aircraft Design and Operations Meeting, p. 3083, 1991. https://doi.org/10.2514/6.1991-3083

[11] Sederberg, T. W., and Farouki, R. T., "Approximation by interval Bézier curves," IEEE Computer Graphics and Applications, no. 5, p. 87-88, 1992. https://doi.org/10.1109/38.156018

[12] Ansys, CFX-18.1, Ansys Inc., 2017.

[13] ANSYS, "Latin Hypercube Sampling Design (LHS)," in Design Exploration User's Guide, ANSYS, Inc., 2013 (Release 15.0), p. 68.

[14] Steven L. Brunton, Bernd R. Noack, and Petros Koumoutsakos, "Machine Learning for Fluid Mechanics," Annual Review of Fluid Mechanics, vol. 52, pp. 477-508, 2020. https://doi.org/10.1146/annurev-fluid-010719-060214

[15] Rajnarayan, D., Haas, A., and Kroo, I., "A multifidelity gradient-free optimization method and application to aerodynamic design," in 12th AIAA/ISSMO multidisciplinary analysis and optimization conference, 2008. https://doi.org/10.2514/6.2008-6020

[16] Kingma, D. P., and Welling, M., "Auto-encoding variational bayes," arXiv preprint arXiv: 1312.6114, 2013. https://doi.org/10.48550/arXiv.1312.6114

[17] Larsen, A. B. L., Sønderby, S. K., Larochelle, H., and Winther, O., "Autoencoding beyond pixels using a learned similarity metric," arXiv preprint arXiv: 1512.09300, 2015. https://doi.org/10.48550/arXiv.1512.09300

[18] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio,Y., "Generative adversarial Networks.," Advances in neural information processing systems, 27, 2014. https://doi.org/10.48550/arXiv.1406.2661

[19] C. Boursier Niutta, E. J. Wehrle, F. Duddeck, and G. Belingardi., "Surrogate Modeling in the Design Optimization of Structures with Discontinuous Responses with Respect to the Design Variables - A New Approach for Crashworthiness Design," Springer International Publishing, Cham, p. 242-258, 2018. https://doi.org/10.1007/978-3-319-67988-4_17

[20] J. Sobieszczanski-Sobieski and R. T. Haftka., "Multidisciplinary aerospace design optimization." survey of recent developments. Structural optimization, vol. 14, no. 1, p. 1-23, 1997. https://doi.org/10.1007/BF01197554

[21] Yuyang Wang, Kenji Shimada and Amir Barati Farimani, "Airfoil GAN: Encoding and Synthesizing Airfoils for Aerodynamic-aware Shape Optimization," arXiv, 2021. https://doi.org/10.48550/arXiv.2101.04757

[22] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, Canada: O'Reilly, 2019.

[23] Mitchell, M., "An introduction to genetic algorithms," MIT Press, 1998. https://doi.org/10.7551/mitpress/3927.001.0001

[24] O. Castillo, P. Melin, J. Kacprzyk,, "Soft Computing for Hybrid IntelligentSystems,," Springer, Berlin, Germany, 2008. https://doi.org/10.1007/978-3-540-70812-4

[25] Skinner SN, Zare-Behtash H., "State-of-the-art in aerodynamic shape optimisation methods." Appl. Soft. Comput. 62, pp. 933-962, 2018. https://doi.org/10.1016/j.asoc.2017.09.030

[26] Khaleghi H, "Stall inception and control in a transonic fan, part A: Rotating stall inception," Aerospace Science and Technology, vol. 41, pp. 250-258, 2015. https://doi.org/10.1016/j.ast.2014.12.004

[27] M. Jalalifar, B. Ghadiri Dehkordi, S. Fallah, "Numerical study of mid-span damper effect on the pattern of flow and operation of transonic compressor blades." Modares Mechanical Engineering (in Persian), vol. 16, no. 10, pp. 218-228, 2016.

http://mme.modares.ac.ir/article-15-9544-en.html

[28] Maas, A. L., Hannun, A. Y., and Ng, A. Y., "Rectifier nonlinearities improve neural network acoustic models." Proc. icml, vol. 30, p. 3, 2013.

[29] Kingma, D. P., and Ba, J., "Adam: A method for stochastic optimization." arXiv preprint arXiv: 1412.6980, 2014. https://doi.org/10.48550/arXiv.1412.6980

[30] Schafer, R. W., et al., "What is a Savitzky-Golay filter," IEEE Signal processing magazine, vol. 28, no. 4, p. 111-117, 2011. https://doi.org/10.1109/MSP.2011.941097

[31] Diamantopoulou, M., Karathanasopoulos, N., & Mohr, D., "Stress-strain response of polymers made through two-photon lithography: micro-scale experiments and neural network modeling." AdditiveManufacturing, p. 102266, 2021. https://doi.org/10.1016/j.addma.2021.102266

[32] Li, J., & Zhang, M., "Data-based approach for wing shape design optimization." Aerospace Science and Technology, 112, p. 106639, 2021. https://doi.org/10.1016/j.ast.2021.106639

[33] Lee, S., & You, D., "Data-driven prediction of unsteady flow over a circular cylinder using deep learning," J. Fluids. Mech. 879, pp. 217-254, 2019. https://doi.org/10.1017/jfm.2019.700

[34] Nie, Z., Jiang, H., & Kara, L. B., "Stress field prediction in cantilevered structures using convolutional neural networks.," in International Design Engineering Technical Conferences and

Computers and Information in Engineering Conference 20, no.1: pp. 011002, 2019. https://doi.org/10.1115/DETC2019-98472

[35] Wang, L., Chan, Y., Ahmed, F., Liu, Z., Zhu, P., & Chen, W., "Deep generative modeling for mechanistic-based learning and design of metamaterial systems." Comput. Methods Appl. Mech. Engrg. 372, p. 113377, 2020. https://doi.org/10.1016/j.cma.2020.113377

[36] LeCun, Y., Bengio, Y., and Hinton, G., "Deep learning," Nature, vol. 521, no. 7553, p. 436, 2015. https://doi.org/10.1038/nature14539

[37] Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., & Bengio, Y., "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation." in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2017. https://doi.org/10.1109/CVPRW.2017.156

[38] Chen, C., & Gu, G. X., "Machine learning for composite materials," MRS Communications 9, no. 2, pp. 556-566, 2019. https://doi.org/10.1557/mrc.2019.32

[39] Saha, S., Gan, Z., Cheng, L., Gao, J., Kafka, O. L., Xie, X., Li, H., Tajdari, M., Kim, H. A., & Liu, W. K., "Hierarchical deep learning neural networks (HiDeNN): An artificial intelligence (AI) framework for computational science and engineering.," Computer Methods in Applied Mechanics and Engineering, 373, p. 113452, 2021. https://doi.org/10.1016/j.cma.2020.113452

[40] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural network," Advances in neural information processing systems, 25, p. 1097-1105, 2012. https://doi.org/10.1145/3065386

[41] Elbadawi, M., Gaisford, S., & Basit, A. W., "Advanced machine-learning techniques in drug discovery." Drug Discovery Today 26, no. 3, pp. 769-777, 2021. https://doi.org/10.1016/j.drudis.2020.12.003

[42] Gao, F., Ren, S., Lin, C., Bai, Y., &Wang, W., "Metamodel-based multi-objective reliable optimisation for front structure of electric vehicle." Automotive Innovation 1, no. 2, pp. 131-139, 2018. https://doi.org/10.1007/s42154-018-0018-8

[43] Tao, S., van Beek, A., Apley, D. W., & Chen, W., "Multi-model Bayesian optimisation for simulation based design." ASME.J.Mech.Des. 143, no. 11, p. 111701, 2021. https://doi.org/10.1115/1.4050738

[44] Sekar, V., Zhang, M., Shu, C., & Khoo, B. C., "Inverse design of airfoil using a deep convolutional neural network." AIAA JOURNAL 57, no. 3, pp. 993-1003, 2019. https://doi.org/10.2514/1.J057894

[45] Wang, Z., Xiao, D., Fang, F., Govindan, R., Pain, C. C., & Guo, Y., "Model identification of reduced order fluid dynamics systems using deep learning.," International Journal for Numerical Methods in Fluids 86, no. 4, pp. 255-268, 2018. https://doi.org/10.1002/fld.4416

[46] Thuereym, N., Weissenow, K., Prantl, L., & Hu, X., "Deep learning methods for Reynolds-Averaged Navier-Stokes simulations of airfoil flows," AIAA JOURNAL 58, no. 1, pp. 25-36, 2020. https://doi.org/10.2514/1.J058291

[47] Oh, S., Jung, Y., Kim, S., Lee, I., & Kang, N., "Deep generative design: Integration of topology optimization and generative models." Journal of Mechanical Design 141, no. 11, 2019. https://doi.org/10.1115/1.4044229

[48] Nie, Z., Lin, T., Jiang, H., & Kara, L. B., "TopologyGAN: Topology optimization using generative adversarial networks based on physical fields over the initial domain." ASME. J. Mech. Des. 143, no. 3, p. 031715, 2021. https://doi.org/10.1115/1.4049533

[49] Zimmerling, C., Poppe, C., & Kärger, L., "Virtual product development using simulation methods and AI.," Lightweight Design worldwide 12, no. 6, pp. 12-19, 2019. https://doi.org/10.1007/s41777-019-0064-x

[50] Voulodimos, A., Doulamis, N., Doulamis, A., and Protopapadakis, E., "Deep learning for computer vision: A brief review," Computational intelligence and neuroscience, vol. 2018, 2018. https://doi.org/10.1155/2018/7068349

[51] Cambria, E., and White, B., "Jumping NLP curves: A review of natural language processing research," IEEE Computational intelligence magazine, vol. 9, no. 2, p. 48-57, 2014. https://doi.org/10.1109/MCI.2014.2307227

[52] Pierson, H. A., and Gashler, M. S., "Deep learning in robotics: a review of recent research," Advanced Robotics, vol. 31, no. 16, p. 821-835, 2017. https://doi.org/10.1080/01691864.2017.1365009

[53] Carrio, A., Sampedro, C., Rodriguez-Ramos, A., and Campoy, P., "A review of deep learning methods and applications for unmanned aerial vehicles," Journal of Sensors, vol. 2017, 2017. https://doi.org/10.1155/2017/3296874

[54] Sobieczky, H., "Geometry generator for CFD and applied aerodynamics," New Design Concepts for High Speed Air Transport, Springer, p. 137-157, 1997. https://doi.org/10.1007/978-3-7091-2658-5_9

[55] Derksen, R., and Rogalsky, T., "Bezier-PARSEC: An optimized aerofoil parameterization for design," Advances in engineering software, vol. 41, no. 7-8, p. 923-930, 2010. https://doi.org/10.1016/j.advengsoft.2010.05.002

[56] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., "Generative adversarial nets," Advances in neural information processing systems, p. 2672-2680, 2014. https://doi.org/10.48550/arXiv.1406.2661

[57] Karras, T., Laine, S., and Aila, T., "A style-based generator architecture for generative adversarial networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4401-4410, 2019. https://doi.org/10.1109/CVPR.2019.00453

[58] Hao, P. Liu, D., Zhang, K., Yuan, Y., Wang, B., Li, G., & Zhang, X., "Intelligent layout design of curvilinearly stiffened panels via deep learning-based method." Materials & Design 197, p. 109180, 2021. https://doi.org/10.1016/j.matdes.2020.109180

[59] Zhou, H., Xu, Q., Nie, Z., & Li, N., "A study on using image-based machine learning methods to develop surrogate models of stamp forming simulations.," Journal of Manufacturing Science and Engineering 144, p. 021012, 2021. https://doi.org/10.1115/1.4051604

[60] Li, J., & Zhang, M., "Efficient aerodynamic shape optimization with deep-learning-based geometric filtering." AIAA JOURNAL 58, no. 10, pp. 4243-4259, 2020. https://doi.org/10.2514/1.J059254

[61] Zhou, H., Xu, Q., Nie, Z., & Li, N., "A study on using image-based machine learning methods to develop surrogate models of stamp formign simulations.," Journal of Manufacturing Science and Engineering 144, p. 021012, 2021. https://doi.org/10.1115/1.4051604

[62] Chang, K., & Cheng, C., "Learning to simulation and design for structural engineering." in International Conference on Machine Learning: pp. 1426-1436, PMLR, 2020.

[63] Pongetti, J., Kipouros, T., Emmanuelli, M., Ahlfeld, R., & Shahpar, S., "Using autoencoders and output consolidation to improved machine learning for turbomachinery applications.," in Proceedings of the ASME 2021 Turbomachinery Technical Conference & Exposition, 2021. https://doi.org/10.1115/GT2021-60158

[64] Jaroslav Menˇc'ık. , "Concise Reliability for Engineers." in Concise Reliability for Engineers, (Chapter 16), April 2016.

[65] Kern S, Müller SD, Hansen N, Büche D, Ocenasek J, Koumoutsakos P. , "Learning Probability Distributions in Continuous Evolutionary Algorithms," A Comparative Review. Nat. Comput. 3, pp. 77-112, 2004. https://doi.org/10.1023/B:NACO.0000023416.59689.4e

[66] Giannakoglou K, Papadimitriou D, Kampolis I., "Aerodynamic shape design using evolutionary algorithms and new gradient-assisted metamodels." Comput. Methods Appl. Mech. Eng. 195, pp. 6312- 6329, 2006. https://doi.org/10.1016/j.cma.2005.12.008

[67] Hamdaoui M, Chaskalovic J, Doncieux S, Sagaut P., "Using multiobjective evolutionary algorithms and data-mining methods to optimize ornithopters' kinematics.," J. Aircraft 47, p. 1504, 2010. https://doi.org/10.2514/1.45046

[68] Van Rees WM, Gazzola M, Koumoutsakos P., "Optimal morphokinematics for undulatory swimmers at intermediate reynolds numbers." J. Fluid Mech. 775, pp. 178-188, 2015. https://doi.org/10.1017/jfm.2015.283

[69] Strom B, Brunton SL, Polagye B., "Intracycle angular velocity control of cross-flow turbines." Nat. Energy 2, pp. 1-9, 2017. https://doi.org/10.1038/nenergy.2017.103

[70] Rezende, D. J., Mohamed, S., and Wierstra, D., "Stochastic backpropagation and approximate inference in deep generative models," arXiv preprint arXiv: 1401.4082, 2014. https://doi.org/10.48550/arXiv.1401.4082

[71] Poon, P. W., and Carter, J. N., "Genetic algorithm crossover operators for ordering applications.," Computers & Operations Research, vol. 22, no. 1, p. 135-147, 1995. https://doi.org/10.1016/0305-0548(93)E0024-N

[72] Voulodimos, A., Doulamis, N., Doulamis, A., and Protopapadakis, E., "Deep learning for computer vision: A brief review." Computational intelligence and neuroscience, 2018. https://doi.org/10.1155/2018/7068349

[73] Cambria, E., and White, B., "Jumping NLP curves: A review of natural language processing research." IEEE Computational intelligence magazine, vol. 9, no. 2, p. 48-57, 2014. https://doi.org/10.1109/MCI.2014.2307227