Structural and Multidisciplinary Optimization (2024) 67:2
https://doi.org/10.1007/s00158-023-03703-9

RESEARCH PAPER

Aerodynamic shape optimization of gas turbines: a deep learning surrogate model approach

Vahid Esfahanian[1] · Mohammad Javad Izadi[1] · Hosein Bashi[1] · Mehran Ansari[1] · Alireza Tavakoli[1] · Mohammad Kordi[1]

Received: 24 July 2023 / Revised: 18 September 2023 / Accepted: 2 November 2023 / Published online: 21 December 2023

Abstract

The improvement of existing turbines requires time-consuming computations that often limit the number of parameters that can be optimized. To address this challenge, this study uses through-flow code, which is about 200 times faster than 3D Computational Fluid Dynamics (CFD), to optimize a two-stage axial turbine with 112 geometrical parameters using a deep learning surrogate model. The surrogate model is a Convolutional Neural Network (CNN) that predicts the flow field and performance indicators of the turbine based on an input database generated by an airfoil generation method and an in-house through-flow code. The surrogate model has two components: a flow field prediction component and a performance prediction component. The network architecture is named conv2D-conv3D, as it employs 2D convolutional layers to map the geometry to the flow field and 3D convolutional layers to extract features from the flow field and output the performance indicators. The optimal hyperparameters of the network are determined by comparing different activation functions, batch sizes, and train-data sizes. The network achieves high accuracy ($R^2 > 0.93$) even with a small fraction of the dataset (30% of data) and it is more than 10^5 times faster than the through-flow code. A vectorized genetic algorithm is applied to the surrogate model to maximize the efficiency and power of the turbine under a constant mass flow rate constraint. The optimization results are validated by through-flow and 3D CFD simulations of the baseline and optimized turbine. The efficiency and power are increased by 0.83% and 1.02%, respectively.

Keywords Gas turbine · Deep learning · Aerodynamic shape optimization · Performance prediction · Convolution neural network

List of symbols

Symbols

$\dot{m}$	Mass flow rate $\left[\frac{kg}{s}\right]$
h	Specific enthalpy $\left[\frac{J}{kg.}\right]$
I	Rothalpy $\left[\frac{J}{kg}\right]$
m	Meridional direction [m]
M	Mach number [−]
S_i	Defining point coordinates [m]
q	Quasi orthogonal direction [m]
V	Absolute velocity $\left[\frac{m}{s}\right]$
R_c	Streamline local radius [m]
s	Specific entropy $\left[\frac{J}{kg..K}\right]$
U	Blade velocity $\left[\frac{m}{s}\right]$
N	Number of blades [−]
P_i	Optional point coordinates [m]
r	Section radius of the airfoil [m]
P	Pressure [Pa]

Subscripts

0	Stagnation condition
1	Leading edge
2	Trailing edge
g	Gauging
tt	Total-to-total

Greek symbols

β	Camber line angle [◦]
ϕ	Meridional slope angle [◦]
γ	Quasi orthogonal angle, stagger angle [◦]

Responsible Editor: Lei Wang

✉ Vahid Esfahanian
evahid@ut.ac.ir

[1] Vehicle, Fuel and Environment Research Institute, School of Mechanical Engineering, College of Engineering, University of Tehran, P.O. Box, Tehran 11155-4563, Iran

$\Delta\beta$ Wedge angle [◦]
η Efficiency [−]
α Flow angle [◦]
ρ Density $\left[\frac{kg}{m^3}\right]$

1 Introduction

Gas turbines have been used widely in power generation, aviation, and oil and gas industries. In recent years, due to global warming, restricted regulations have been imposed to reduce greenhouse gases (Wagner et al. 2015). Additionally, increasing the turbine efficiency by 1–2% would result in hundreds of millions of dollars of savings in a year (High-Efficiency Gas Turbines Will Play Growing Role in the Energy Transition, 2018; Osseyran and Giles 2015). These two and many other reasons motivated researchers to optimize available turbines.

In this context, there are three main methods of turbine optimization. These methods are based on flow solvers, which are 1D (mean-line), 2D (streamline curvature), and 3D solvers. Stochastic optimization algorithms are very time-consuming and require the calculation of objective functions at many points. Turbine aerodynamic calculation takes from seconds in 1D to hours in 3D solvers. Therefore, it is not reasonable and sometimes possible to use real functions in optimization algorithms. Surrogate models make a good trade-off between computation cost and accuracy of function estimation (Li and Zheng 2017).

In general, there are two types of surrogate models, namely black box and projection-based methods. The black box method directly constructs mapping between inputs(s) and output(s). The projection-based method reduces the dimensionality of the governing equations by projecting them to a subspace (Li and Zhang 2022). Various surrogate models have been implemented by researchers using these two methods. Response surface (Martin et al. 2019), kriging (Raul and Leifsson 2021; Mohammadi-Ahmar et al. 2022), Support Vector Regression (SVR) (Jia et al. 2022; Hu et al. 2022), and Gaussian process (Luo et al. 2023; Catalani et al. 2023) are categorized as black box methods, while the reduced-order method (Eivazi et al. 2020; Ashouri et al. 2022) is a projection-based method subset.

A large number of Computational Fluid Dynamics (CFD) calculations is needed for training the surrogate model. There are many high-dimensional data such as flow field contours in CFD calculations, which can be useful for the surrogate model to predict output parameters. However, the black box method does not utilize them (Liu 2021), which compromises the generality of surrogate models.

Thanks to the improvements in hardware such as Graphics Processing Units (GPU) at affordable prices and the availability of labeled data, deep neural networks attracted the attention of many researchers as a powerful surrogate model. Ates and Gorguluarslan (Ates and Gorguluarslan 2021) presented a two-stage convolutional encoder-decoder network model for topology optimization that aims to minimize structural disconnections and pixel-wise errors. The model is tested on both 2D and 3D datasets and shows improved performance over a single network. Lui et al. (Lui et al. 2022) developed a Deep learning surrogate model for acoustic particle patterning. They used a physics-informed encoder–decoder model. Duru et al. (Duru et al. 2022) used a deep learning model to predict the flow field around airfoils in transonic conditions ($M = 0.7$). The model was trained on a dataset of CFD simulations and was able to predict the effects of angle of attack and airfoil shape variations on the flow field. A model with the capability to give a solution for various angles of attack, Reynolds numbers, and Mach numbers was proposed by Chen et al. (Chen and Thuerey 2023). Zhao et al. (Zhao et al. 2021) developed a novel deep surrogate model for heat source layout temperature field prediction using feature pyramid networks, pairwise temperature field difference, and deep transfer learning. The flow and heat transfer of U-bend channels were predicted by a deep learning method used by Wang et al. (Wang, et al. 2022). The approaches presented in these papers show that deep learning is able to predict the flow field with a high degree of accuracy.

The application of flow field prediction in turbomachinery has been studied in many studies. Shi et al. (Shi et al. 2020) obtained the off-design performance of a S-CO_2 turbine using two networks, flow prediction and performance prediction, which led to relative errors of $\pm 1\%$ and $\pm 5\%$ for efficiency and power, respectively. (Li, et al. 2023) proposed a fast Multidisclinary prediction model for gas turbines based on supervised graph learning approaches. They used an Aerodynamic Strength Prediction Graph neural network to predict the temperature field, aerodynamic performance, and strength performance of a gas turbine blade with complex cooling channels under different boundary conditions. (Feng et al. 2023) developed a physics-informed deep learning model for predicting cascade loss and flow field in aero-engine compressor design. They showed that their model outperformed empirical and simple deep learning models by incorporating the velocity distribution in the boundary layer as an intermediate variable. (Du 2022) optimized a single-stage axial turbine with seven design parameters using a double convolutional neural network and automatic differentiation in the neural network. They performed 1000 CFD cases, which took 500 h in total. (Wang et al. 2021a) proposed a Dual-CNN model for the prediction and reconstruction of the aerodynamic performance and fields of aero-engine turbines under multiple conditions. A multi-objective optimization

of efficiency and torque with eight design parameters is also performed by the model by automatic differentiation, and a Pareto solution was obtained.

The deep learning method is an accurate and fast method for predicting the flow fields and performance of turbomachines, as mentioned. The attention of most researchers has been focused on three-dimensional flow calculations and very few parameters; however, the goal of this research is to optimize 112 geometric parameters of a two-stage axial turbine using a new two-component deep learning architecture. The high number of variables makes the production of a suitable amount of high-fidelity training data such as 3D CFD very time-consuming, expensive, and to some extent impossible. Therefore, the streamline curvature method, which has appropriate accuracy and low computational cost, is used for making the training data set. Automatic differentiation, which is fast but does not exploit the full potential of neural networks offered by vectorization, has been utilized by some studies. Gradient-based methods are susceptible to local optimization, requiring multiple initial solutions to be performed by the user for the optimization. A vectorized genetic algorithm is used to address these issues, which combines the high performance of the deep learning surrogate model with the robust search ability of the genetic algorithm.

2 Numerical procedure

2.1 Streamline curvature

Through-flow methods which are fast and reasonably accurate are widely used in the industry (Chaquet et al. 2017). Among different types of through-flow models, the streamline curvature (SLC) method has a better performance in gas turbines, especially in transonic conditions, due to its greater stability (Tiwari et al. 2013). In the SLC method, it is assumed that the flow does not change in the circumferential direction, and this assumption results in solving the two-dimensional flow in the meridional plane, which is a compromise between speed and accuracy in one-dimensional and three-dimensional models (Persico and Rebay 2012). In this method, momentum and energy conservation equations are considered in intrinsic coordinates, i.e. streamlines. Furthermore, since the calculation of the losses caused by the movement of the flow between the blades requires very precise details, and it is almost impossible to compute them analytically, empirical models are used to estimate the losses in the turbine. In addition to the above equations, the momentum equation in the spanwise direction must also be established. In this equation, the curvature of streamlines is also considered; this is why this method is called streamline curvature (Aungier 2006). The radial equilibrium equation is as follows:

$$V_m \frac{\partial V_m}{\partial q} = A + BV_m^2 \tag{1}$$

$$A = \cos^2(\alpha)\left[\frac{\partial I}{\partial q} - T\frac{\partial s}{\partial q}\right] \tag{2}$$

$$\begin{aligned} B = \cos^2(\alpha)\Bigg[&\frac{\cos(\phi-\gamma)}{R_c} - \frac{\tan(\alpha)}{\cos^2(\alpha)}\frac{\partial \alpha}{\partial \mathrm{q}} \\ &+ \frac{2U}{V_{\mathrm{m}}}\tan(\alpha)\frac{\cos(\gamma)}{r} \\ &- \tan^2(\alpha)\frac{\cos(\gamma)}{r} + \frac{\sin(\phi-\gamma)}{V_m}\frac{\partial V_m}{\partial m}\Bigg] \end{aligned} \tag{3}$$

Solving this equation for a Quasi Orthogonal (QO) needs the meridional velocity in the next QO to compute $\partial V_m/\partial m$. Novak (Novak 1967) used the continuity equation to obtain the following relationship that converts the radial equilibrium equation from a PDE to an ODE, which can be solved explicitly:

$$\frac{\sin\phi}{C_m}\frac{\partial C_m}{\partial m} = -\frac{\left(1 + M_\theta^2 + \frac{r}{r_m\cos\phi}\right)\frac{\sin^2\phi}{r} + \tan\phi\frac{\partial\phi}{\partial r}}{1 - M_m^2} \tag{4}$$

The intersection of the streamlines with the QO (the lines that are almost perpendicular to the streamlines and are chosen arbitrarily and remain fixed) forms the computing stations as shown in Fig. 1. In addition to the applying of the momentum and energy equations between the stations located on a streamline and the applying of the radial equilibrium equation on the stations located in a QO, it is necessary that the continuity equation also be satisfied for all QOs:

$$\dot{m} = 2\pi \int_0^{y_s} r\rho V_m \cos(\phi-\gamma)dy \tag{5}$$

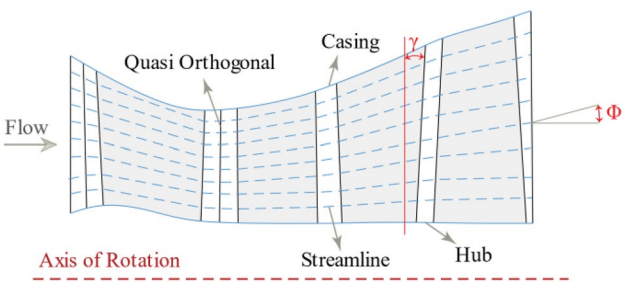

Fig. 1 Schematic of calculating stations

Table 1 NASA turbine specifications

Variable Name	Value
Stage no.	2
Blade no. first stator	50
Blade no. first rotor	61
Blade no. second stator	39
Blade no. second rotor	52
The radius of the mean line (mm)	330.2

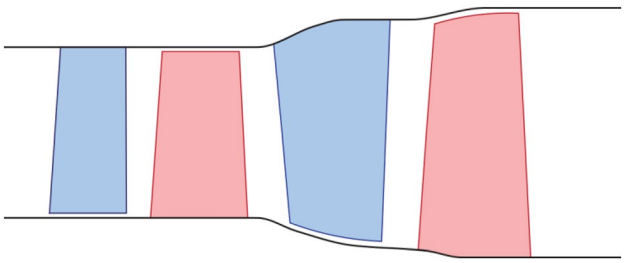

Fig. 2 Meridional view of NASA turbine

Table 2 NASA turbine operating conditions

Variable Name	Value
Total inlet pressure (Pa)	101,350
Total inlet temperature (K)	378
Total pressure ratio	1.2–4
Non-dimensional rotational speed (rpm)	4407
Maximum efficiency	93.2

It should be noted that the continuity equation must be established between each two adjacent streamlines in addition to the whole QO from hub to casing. Since the location of the streamlines is not known at the beginning of the solution, first these lines are considered with an equal distance from the hub to the tip, and when the entire flow is solved, they are rearranged in such a way that the flow rate passing through two streamlines is constant from the inlet to outlet of the turbine. After the repositioning of streamlines, the flow solution is recalculated until the location of the streamlines is converged. The number of streamlines and QOs is arbitrary and is determined by the user, but it should be in a way such that at least one QO is placed in the inlet and outlet of each row. On the other hand, it should not be a large number, which causes numerical instability (Denton 1992). For this study, the number of streamlines and QOs are considered as 12 and 9, respectively.

2.2 Streamline curvature validation

In this research, the experimental data of a NASA turbine conducted by Whitney et al. (Whitney et al. 1972) are used to validate the SLC code and then optimize its geometry. The main specifications and meridional view of this turbine are listed in Table 1 and Fig. 2, respectively:

The NASA turbine boundary conditions are given in Table 2. Using the existing SLC code, the turbine map is extracted and compared with the experimental data. Figure 3 shows this comparison for efficiency and non-dimensional mass flow rate (NDM). According to Fig. 3, at the working pressure of the turbine (Total pressure ratio of 3.2), the

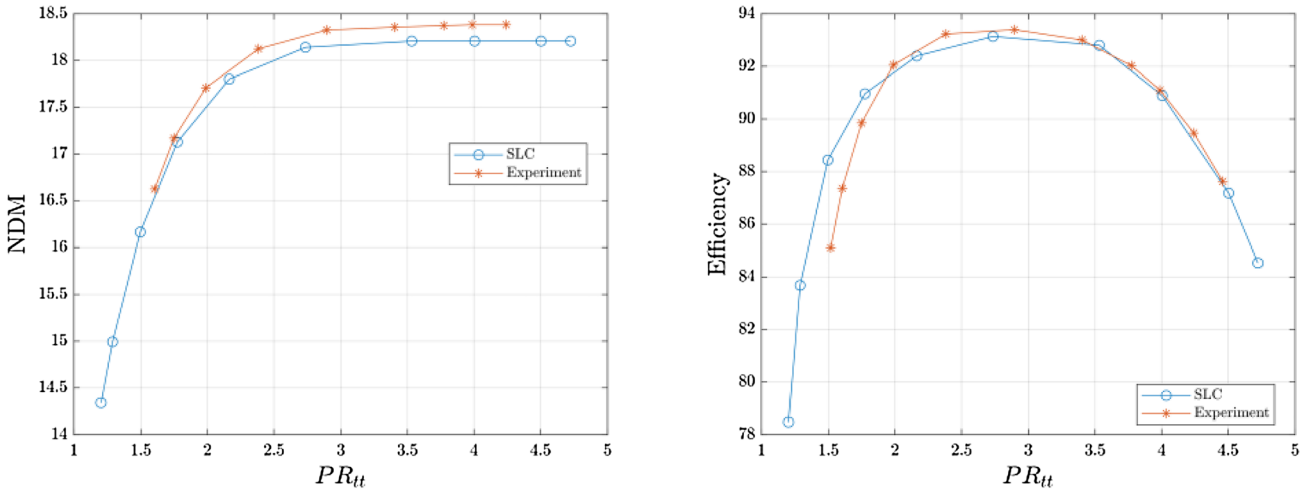

Fig. 3 Comparison of experimental results and SLC solver

difference between the efficiency and the dimensionless mass flow rate resulting from the simulation and the experimental mode is less than 0.25%.

2.3 Airfoil generation and matching method

Reference (Novak 1967) reported the pointwise geometry of airfoils in three spans: hub, mean-line, and tip, except for the leading and trailing edges. This data is not sufficient to be used in SLC code because detailed information on blades, such as the entry and exit angle, chord, maximum thickness, trailing edge thickness, etc. is needed. On the other hand, during the optimization process, combinations of different design parameters lead to geometries that are not feasible, as is be discussed in Sect. 3.2. To answer these two needs, a blade generation method (Tiwari et al. 2013) is used.

This method represents an airfoil with 11 geometrical parameters, as shown in Fig. 4, which are:

- Radius of leading edge (R_1)
- Camber line angle of the leading edge (β_1)
- Wedge angle of the leading edge ($\Delta\beta_1$)
- Radius of the trailing edge (R_2)
- Camber line angle of the trailing edge (β_2)
- Wedge angle of the trailing edge ($\Delta\beta_2$)
- Stagger angle (γ)
- Gauging angle (β_g)

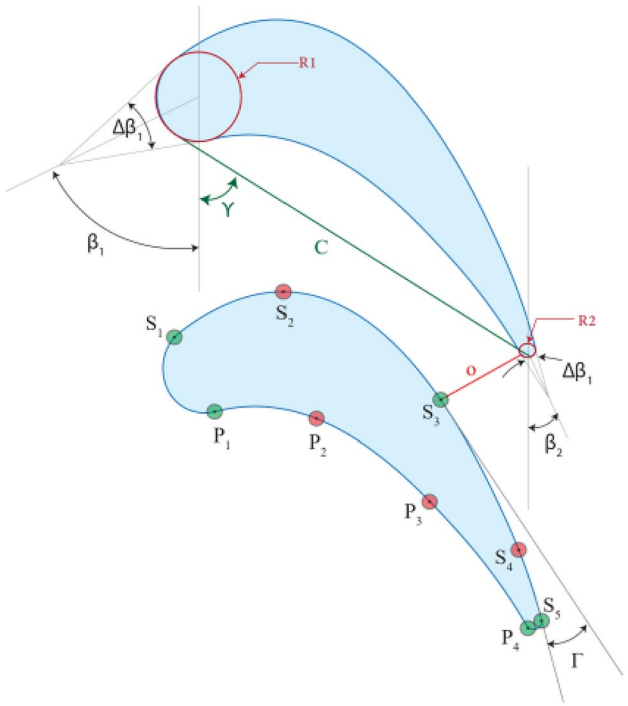

Fig. 4 Schematic of airfoil generation

- Chord (C)
- Number of blades (N)
- Section radius of the airfoil (r)

Given these parameters, three points on the suction side (S_1, S_3, and S_5) and two points on the pressure side (S_2 and S_4) can be determined. These points are used to construct the suction and pressure side curves using spline interpolation. The leading and trailing edges are circular arcs that are tangent to the suction and pressure side curves.

To obtain the variables of each airfoil, a trial-and-error method is inefficient and inaccurate. Therefore, a genetic algorithm method is employed to generate and evaluate several combinations of input variables. Among the 11 variables for each airfoil, only 9 are unknown, as the number of blades and their section radius are given. The objective function to be minimized is the sum of the absolute distances between the generated airfoil points and the reference points. After obtaining the parametrized information of the 4 blades of the desired turbine (12 airfoils in total), generation of various feasible geometries is possible.

2.4 Design of experiment

The meridional view of the turbine is fixed in this research, so the section radius of the airfoil (r) is constant. Thus, each airfoil has 10 parameters out of 11. In addition, the airfoils of a blade share the same number of blades (N). Hence, each blade, which consists of three airfoils, has 28 geometrical variables. The turbine in this research has two stages with four blades, so there are 112 variables in total. The turbine is well-designed and has relatively high efficiency, so it is assumed that only minor changes are needed to improve the performance. Therefore, the optimization process varies the angles by ±4 degrees and the geometric dimensions (chord, leading and trailing edge radius, and the number of blades) by ±5%.

The neural network requires a large number of training data to establish an accurate relation between input(s) and output(s). Therefore, a design of experiment (DOE) method is used to create an input dataset that covers a diverse range of different geometries. The diversity of training data directly affects the accuracy and comprehensiveness of the surrogate model. Several methods for the design of experiments have been used by researchers, such as the full factorial method (Karthikeyan, et al. 2019), Latin hypercube (Misaka 2020), and Sobol (Karimi 2021). The Sobol method (Sobol 1967) has the advantage of covering the variable space of the problem and reproducing data. Hence, Using the Sobol method, 60,000 different conditions are generated. Out of these conditions, 12,400 cases are deemed feasible by the airfoil generation code and run by the SLC code to obtain the turbine flow field solution. The geometrical parameters

of feasible and non-feasible conditions are stored and used in Sect. 3.2. About 15% of cases had a choking condition. At each simulation step, relative Mach number, static pressure, and entropy contours as well as mass flow rate, power, and efficiency are extracted. Figure 5 and Fig. 6 show the correlation between input variables and efficiency and mass flow rate, respectively.

3 Structure of neural network

3.1 Conv2D-Conv3D

As mentioned, in this research, flow field information including relative Mach number, static pressure, and entropy fields will be used to estimate the characteristics of the turbine. For this purpose, the conv2D-conv3D network comprises two neural networks, namely flow field prediction and

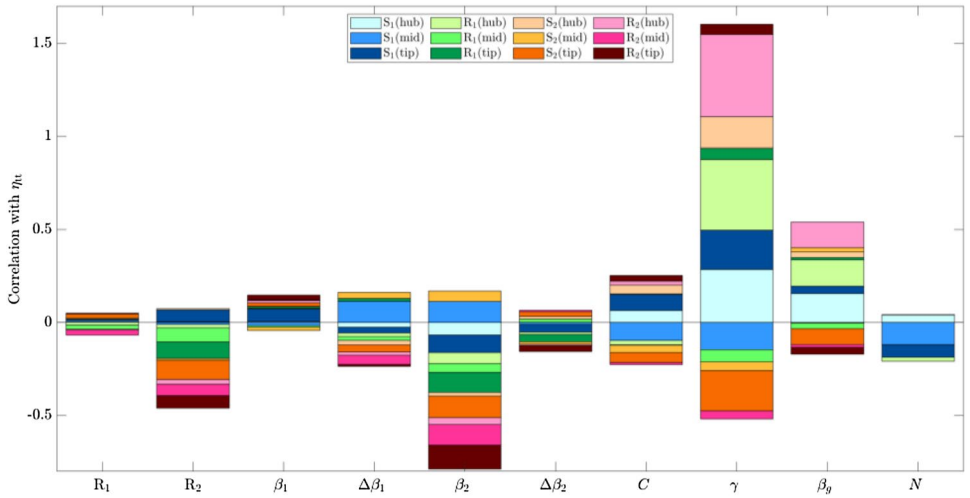

Fig. 5 Correlation plot of input variables and efficiency

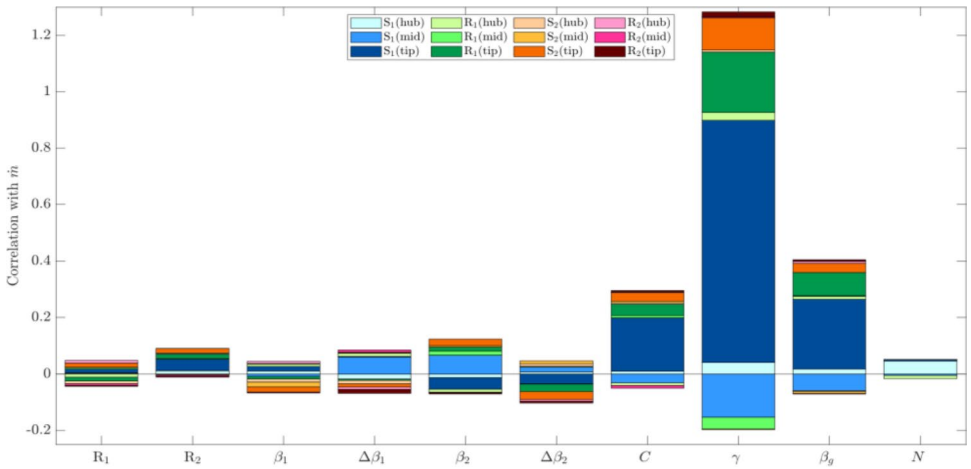

Fig. 6 Correlation plot of input variables and mass flow rate

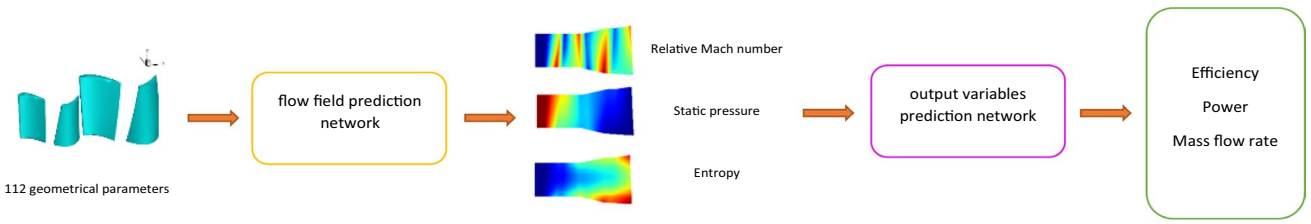

Fig. 7 Overall structure of the conv2D-conv3D network

performance prediction network is used. According to Fig. 7, the geometrical inputs are given to the flow field prediction network to obtain the relative Mach number, static pressure, and entropy fields. The output of the flow field prediction network is the input of the performance prediction network, which estimates the efficiency, power, and mass flow rate values.

the flow field prediction network maps the input to the mentioned flow fields. According to Sect. 2.3, there are 112 geometrical variables that are given as a vector to the input of the flow field prediction network. According to Sect. 2.1, each of the contours of the flow field is a 9×12 matrix (9 QOs and 12 streamlines). Therefore, the flow field prediction network maps a one-dimensional vector (112×1) to a two-dimensional matrix (9×12). The input vector is first connected to a dense layer with 108 neurons, then it is reshaped into a 9×12 matrix using the reshape layer. From here, the network is divided into three structurally identical branches, each of which contains several 2D convolution layers, followed by an averaging layer, and finally, another 2D convolution layer. Each of these branches predicts a flow contour. Figure 8 shows the proposed architecture of the flow field prediction.

In the proposed architecture, three branches are incorporated, each consisting of six layers with 64 filters per layer. This specific configuration was arrived at through a process of trial and error.

A merging layer is also introduced in the architecture, which serves to consolidate the outputs from each branch. Three strategies were tested for this layer, namely averaging, minimum, and maximum. Upon evaluation, it was found that the averaging operation yielded the highest accuracy.

Figure 9 shows the comparison of three activation functions, two batch sizes, and five train-to-test split ratios based on R^2. According to Fig. 9 case (0.9, GELU, 64), which means using 0.9 percent of data as train data,

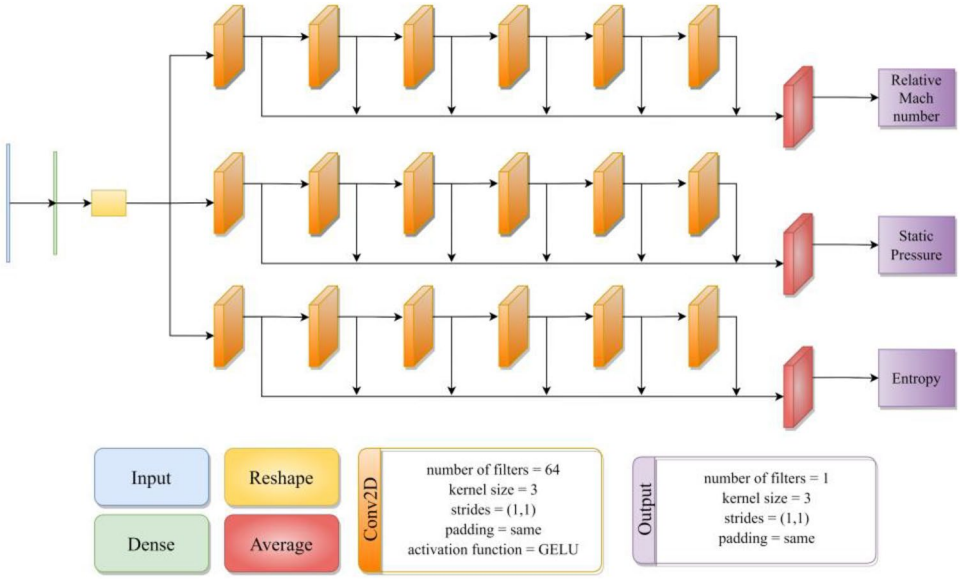

Fig. 8 Architecture of the proposed flow field prediction network

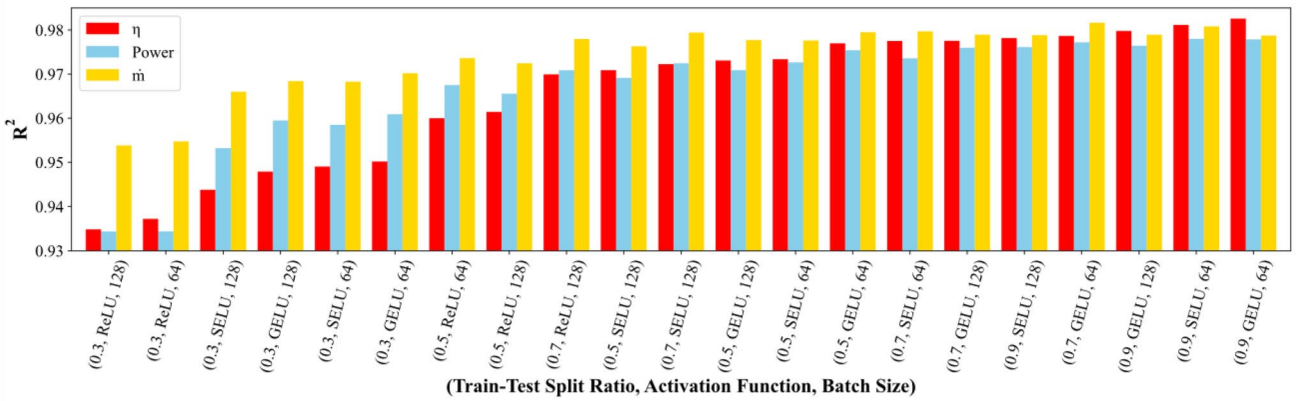

Fig. 9 The flow fieled prediction network hyperparameter tuning

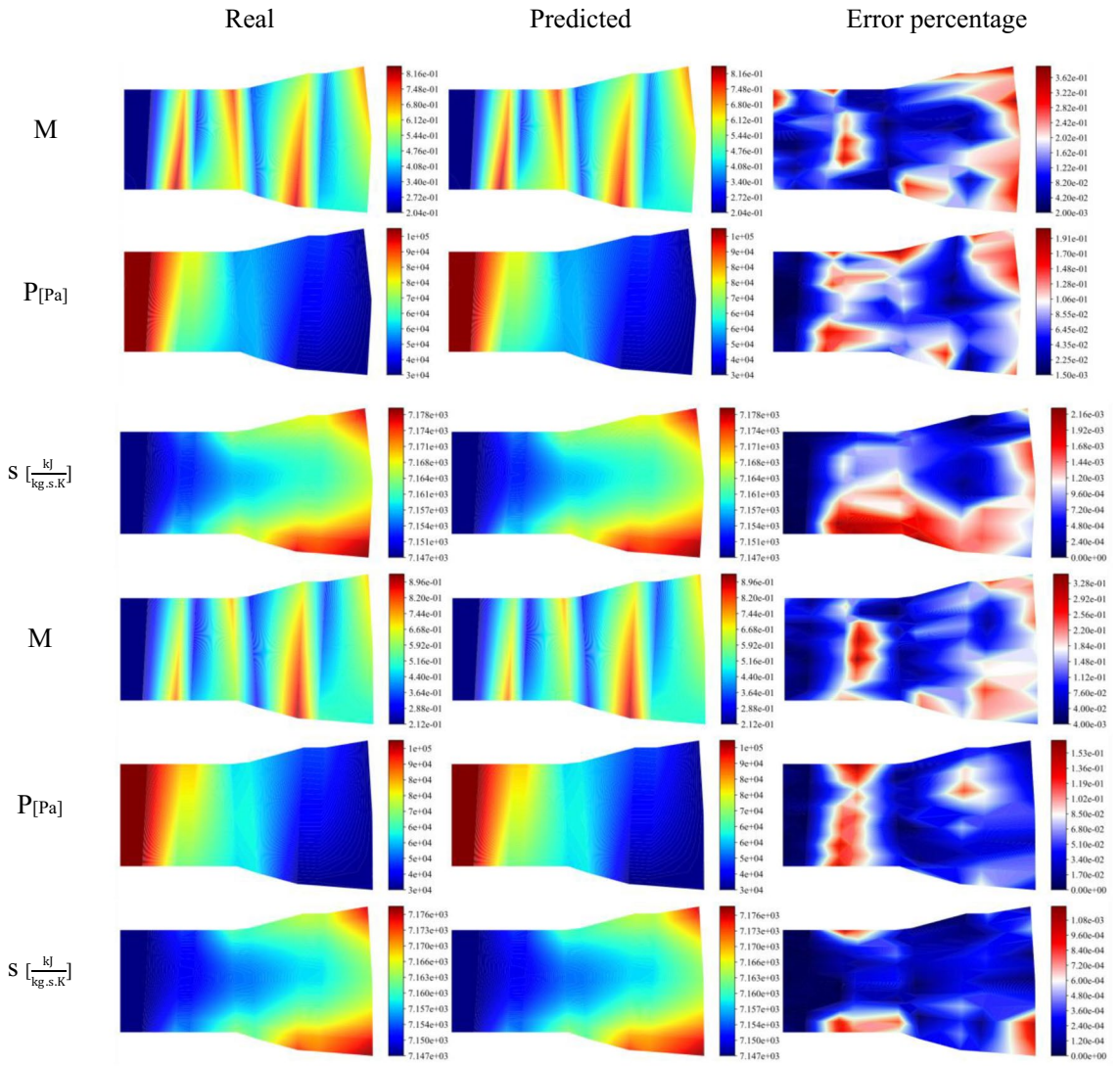

Fig. 10 Comparison of real and predicted flow fields

Gaussian Error Linear Units (GELU) (Hendrycks 2016) as activation function, and batch size of 64 leads to better accuracy. Even with a relatively small dataset, good accuracy is achievable. A reasonable recommendation in the literature for a very sparse dataset is transfer learning (Wang et al. 2021b; Wang et al. 2023).

The comparison between real and predicted flow fields is depicted in Fig. 10:

The second network, which is related to the estimation of efficiency, power, and mass flow rate of the turbine, receives the relative Mach number, static pressure, and entropy matrices from the SLC output. Then, a set of features is extracted

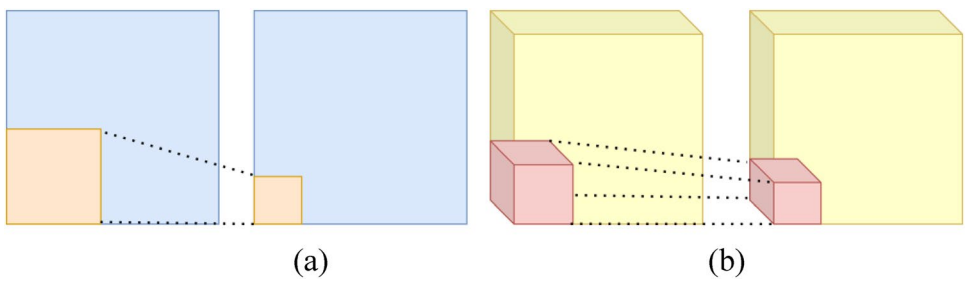

Fig. 11 A comparison of how **a** 2D convolution and **b** 3D convolution operate

from the flow fields through a series of 3D convolution layers. The 3D convolution layer is used in this study because it can capture the spatial and temporal dependencies in the 3D input data, such as the flow field of the turbine. By applying sliding cuboidal convolution filters to the 3D input data, the 3D convolution layer can extract features from each contour and the other contours in the depth dimension, as shown in Fig. 11. The 3D convolution layer can also exploit the information in 3D dimension more effectively than 2D convolution, which only traverses in 2D when the feature map's depth is the same as the kernel's depth. Therefore, the 3D convolution layer is chosen as a suitable method for predicting and reconstructing the aerodynamic performance from the flow fields of the turbine. After conv3D layers, a 3D global average pooling layer is used, and finally, using two dense layers and an output layer, the values of efficiency, power, and mass flow rate are obtained. The architecture of this network is presented in Fig. 12.

Adam optimizer (Kingma and Ba 1412) and Mean Absolute Error (MAE) loss function are used in both networks. As illustrated in Fig. 13, overfitting is checked using the callback function, and learning is stopped if this criterion is violated. The learning rate is initially 0.001, and if the validation loss is not reduced within 25 epochs, the learning rate is reduced by a factor of 0.9 using the ReduceLROnPlateau function. Therefore, unlike other methods that reduce the learning rate after a specific number of epochs, this callback reduces the learning rate only if there is no improvement in a selected metric. Also, both networks are trained by NVIDIA Tesla T4 provided by Google Colab.

After training both networks and combining them together that forms the conv2D-conv3D network, a speed comparison between the SLC simulation and surrogate model is done, which is displayed in Fig. 14. This figure shows that the surrogate model is about 500 times faster when only one design point is calculated. Multiple design points can also be evaluated simultaneously by the surrogate model, owing to the vectorization ability of neural networks. This makes it up to 10^5 times faster

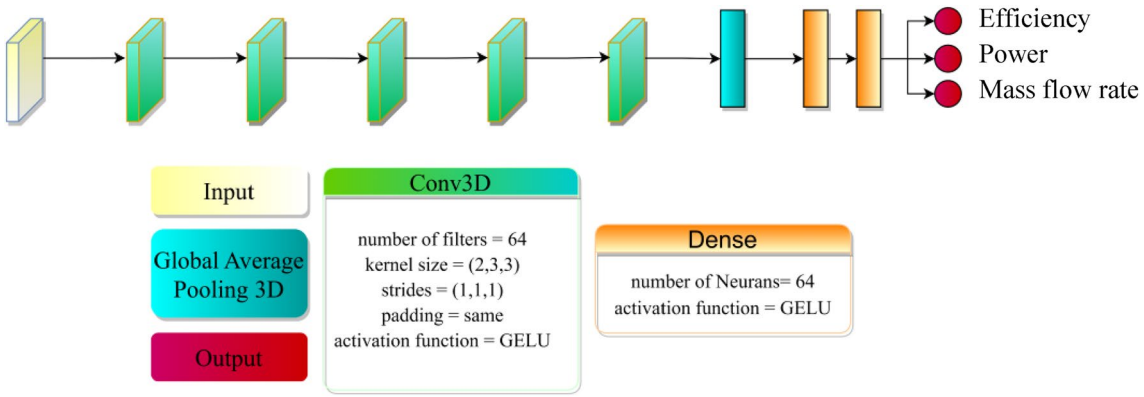

Fig. 12 Architecture of the proposed performance prediction network

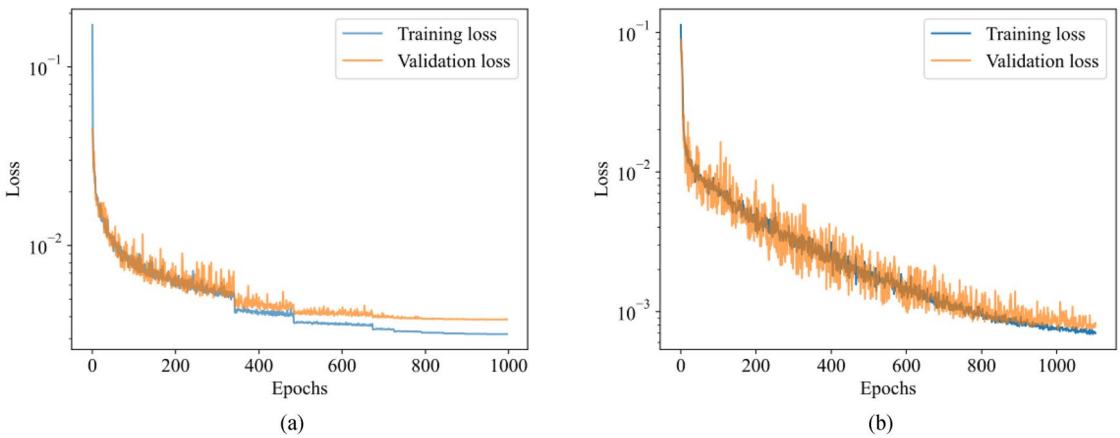

Fig. 13 Convergence of network loss: **a** filed prediction network, **b** performance prediction network

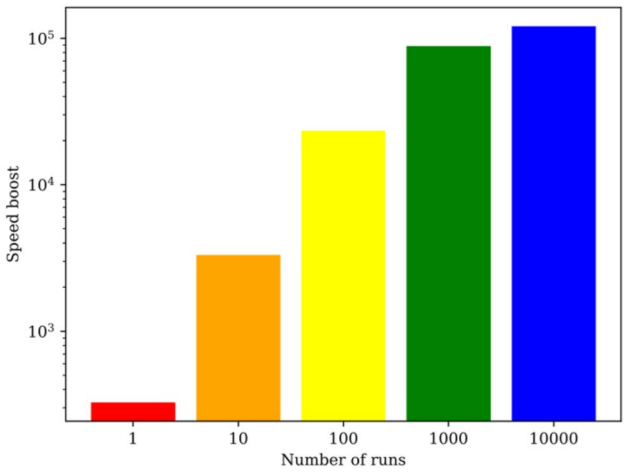

Fig. 14 Speed boost gained by the surrogate model

than evaluating each design point separately. This ability enables a vectorized genetic algorithm that evaluates an entire generation at once.

3.2 Airfoil classifier

The airfoil generation code can distinguish feasible and non-feasible cases in the optimization process. The feasibility criteria are: no more than one sign change of concavity and slope of thickness and no throat at any location other than point S_3 in Fig. 4, which is determined by the inlet parameters of the blade. If these conditions are violated, the code rejects the blade design as non-feasible. However, this code cannot do vectorized calculations and slows down the optimization process. Therefore, a neural network is made to classify airfoils using 112 geometrical parameters. A network with 3 dense layers of 12, 14, and 28 neurons each and GELU activation function, and an output layer of 1 neuron and sigmoid activation function is found to be the optimal configuration by random search. The network is trained for 200 epochs with a batch size of 256, using Adam optimizer, binary cross entropy loss, and early stopping callback. This network achieved R^2 of 0.955 for test and 0.972 for training.

4 Optimization

This research uses a vectorized genetic algorithm to optimize the efficiency of the blades under the constraints of constant flow rate (with a maximum change of 1%), no power reduction, and blade feasibility. The constant flow rate constraint ensures that the thermodynamic cycle of the gas turbine remains unchanged. Otherwise, the optimization might result in drastic changes in the mass flow rate and the cycle, requiring a modification of the compressor to match the new cycle. To avoid this problem, the mass flow rate is kept constant.

Table 3 Genetic algorithm variables

Variable Name	Value
Initial population	1000
Number of generations	200
Mutation rate	0.1
Elitism ratio	0.01
Maximum generation without improvement	10
Selection type	Tournament

The optimization process with the genetic algorithm requires defining some variables, which are shown in Table 3.

Figure 15 illustrates the optimization process using the vectorized genetic algorithm. First, an initial population of 1000 individuals with 112 parameters each is generated and represented as a 1000×112 matrix. This matrix is then passed to the Conv2d-Conv3d networks and the airfoil classifier. The Conv2d-Conv3d networks produce a 1000×3 matrix of output parameters (efficiency, power, and mass flow rate). The airfoil classifier uses the initial population to remove non-feasible solutions by penalizing their efficiency. The mass flow rate criterion is evaluated next by comparing the mass flow rates obtained from the individuals with the base mass flow rate ($\dot{m}_{base}$). The individuals are penalized if their mass flow rates deviate from the defined range of $\pm 1\%$ change. In the subsequent step, the individuals with lower power than the base design (P_{base}) are penalized. Then, the normal genetic algorithm process is applied, and a new generation is formed by selection, crossover, and mutation. This new generation is reevaluated by the objective function described above unless the stopping criteria are satisfied.

After 134 generations of optimization, the process converged. The solution was verified with the SLC solver to ensure that it is indeed better than the base solution and not affected by possible inaccuracies of the neural network. The verification results are shown in Table 4.

The final changes to the geometry are shown in Fig. 16:

4.1 Validation of optimal solution with CFD

To verify the optimization results, 3D, steady, viscous, compressible, and turbulent Reynolds-averaged Navier–Stokes (RANS) equations are solved using Ansys CFX. The *k–ω SST* model (Salviano et al. 2021) is selected for the turbulence. The boundary condition is the total pressure and temperature of the inlet and the static pressure of the outlet (corresponding to a total pressure ratio of 3.2). The total pressure and temperature of the inlet and the static pressure

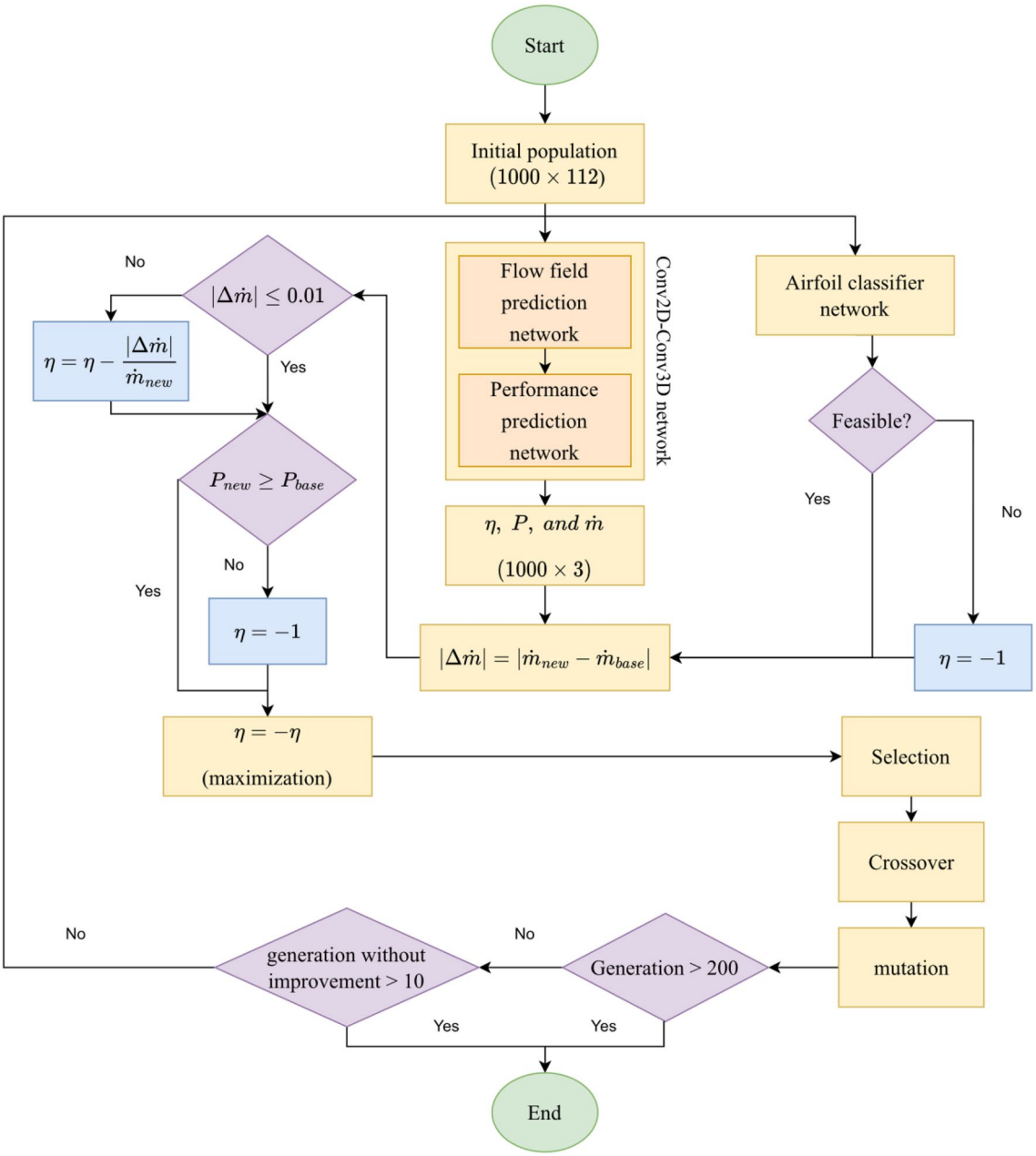

Fig. 15 The objective function flowchart

Table 4 Comparison of the baseline and optimal state

	Efficiency	Power [kW]	Mass flow rate $\left[\frac{\text{kg}}{\text{s}}\right]$
Baseline	93.132	1.468	16.279
Neural network prediction for optimal state	93.762	1.475	16.289
Optimal mode with SLC solver	93.907	1.483	16.336

of the outlet (corresponding to the total pressure ratio of 3.2) are used as the boundary conditions. Figure 17 shows the computational domain gridded by Ansys TurboGrid software, which creates structured grids for turbomachines without user blocking.

To check the grid independence, the efficiency is calculated and compared in several different grids. According to Fig. 18, in case 3, the solution accuracy and the computational cost are well balanced. Therefore, this case is chosen as the final solution.

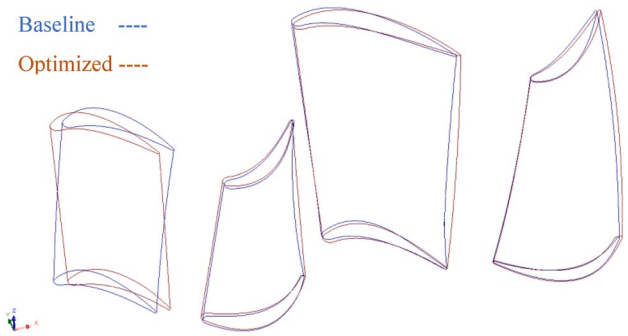

Fig. 16 Comparison of baseline and optimized geometries

Fig. 17 Computational grid of NASA two-stage turbine

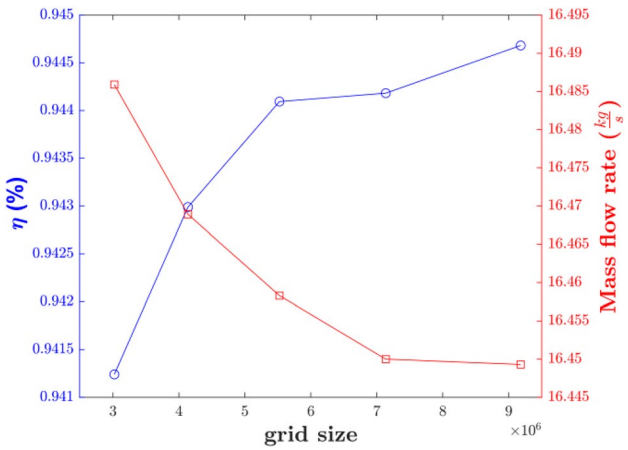

Fig. 18 Grid independence study

The comparison of the baseline and optimal mode in the CFX software shows that the efficiency is improved by 0.3 percent, confirming the optimization result. The difference between the SLC and 3D results might arise from the turbulence modeling in the 3D RANS and the experimental correlations in the SLC for loss predictions. However, both solvers predict an improvement of the baseline geometry. A notable point is that a 3D CFD calculation requires about 30 min with 4 CPU cores (CPU: Core i7-4720HQ), whereas SLC can perform about 200 calculations with the same time and hardware.

5 Conclusions

This research uses deep learning to optimize the efficiency and power of a two-stage axial turbine with 112 geometrical parameters under a constant mass flow rate constraint. The optimization process involves the following steps:

1. Creating a dataset of 12, 600 samples using SLC simulations.
2. Training two deep neural networks to predict the flow field and performance of the turbine, respectively, and combining them into a unified surrogate model (conv2D-conv3D).
3. Applying a vectorized genetic algorithm to the surrogate model.

The main findings are as follows:

- The GELU activation function outperformed the SELU and ReLU functions.
- The Conv3D layer effectively captured spatial and temporal dependencies in 3D input data (e.g., flow field) by using sliding cuboidal convolution filters.
- The deep learning model achieved high accuracy ($R^2 > 0.93$) even with a small fraction of the dataset (30% of data).
- Compared to SLC simulations, which are about 200 times faster than 3D CFD simulations, the deep learning model achieves a speedup of more than 10^5.
- The combination of a deep learning surrogate model and vectorized genetic algorithm resulted in fast and accurate optimization.
- Optimization increased the efficiency and power by 0.83% and 1.02%, respectively.

Funding There is no funding source.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Replication of results This paper has provided all the essential information to obtain the turbine optimization data and the deep learning surrogate model that can reproduce the results. The results can be replicated by following this information. The Python code to reproduce the results, including the conv2D-conv3D network, is also available from the corresponding author upon reasonable request.

References

Ashouri M, Khaleghian S, Emami A (2022) Reduced-order modeling of conductive polymer pressure sensors using finite element simulations and deep neural networks. Struct Multidisc Optim 65(5):146

Ates GC, Gorguluarslan RM (2021) Two-stage convolutional encoder-decoder network to improve the performance and reliability of deep learning models for topology optimization. Struct Multidisc Optim 63(4):1927–1950

Aungier RH (2006) Turbine aerodynamics: axial-Flow and Radial-Flow turbine design and analysis. In: ASME Press eBooks. https://doi.org/10.1115/1.802418

Catalani G, Costero D, Bauerheim M, Zampieri L, Chapin V, Gourdain N, Baqué P (2023) A comparative study of learning techniques for the compressible aerodynamics over a transonic RAE2822 airfoil. Comput Fluids 251:105759

Chaquet JM, Corral R, Fernandez A (2017) Accurate method to reproduce throughflow results with a meanline solver. in turbo expo: power for land, sea, and air. American Society of Mechanical Engineers, New York

Chen L-W, Thuerey N (2023) Towards high-accuracy deep learning inference of compressible flows over aerofoils. Comput Fluids 250:105707

Denton JD (1992) The calculation of three-dimensional viscous flow through multistage turbomachines. J Turbomach 114(1):18–26

Du Q et al (2022) Performance prediction and design optimization of turbine blade profile with deep learning method. Energy 254:124351

Duru C, Alemdar H, Baran OU (2022) A deep learning approach for the transonic flow field predictions around airfoils. Comput Fluids 236:105312

Eivazi H, Veisi H, Naderi MH, Esfahanian V (2020) Deep neural networks for nonlinear model order reduction of unsteady flows. Phys Fluids 32(10):105104

Feng Y, Song X, Yuan W, Lu H (2023) Physics-informed deep learning cascade loss model. Aerosp Sci Technol 134:108165

Hendrycks, D. and K. Gimpel, *Bridging nonlinearities and stochastic regularizers with gaussian error linear units*. 2016.

High-Efficiency Gas Turbines Will Play a Growing Role in the Energy Transition. 2018; Available from: https://www.ge.com/power/transform/article.transform.articles.2018.sep.high-efficiency-gas-turbines.

Hu H, Song Y, Yu J, Liu Y, Chen F (2022) The application of support vector regression and virtual sample generation technique in the optimization design of transonic compressor. Aerosp Sci Technol 130:107814

Jia R, Xia H, Zhang S, Su W, Xu S (2022) Optimal design of Savonius wind turbine blade based on support vector regression surrogate model and modified flower pollination algorithm. Energy Convers Manag 270:116247

Karimi MS, Raisee M, Salehi S, Hendrick P, Nourbakhsh A (2021) Robust optimization of the NASA C3X gas turbine vane under uncertain operational conditions. Int J Heat Mass Transf 164:120537

Karthikeyan T, Avital E, Nithya V, Abdus S (2019) Optimization of a horizontal axis marine current turbine via surrogate models. Ocean Syst Eng. https://doi.org/10.12989/ose.2019.9.2.111

Kingma, D.P. and J. Ba, *Adam: A method for stochastic optimization.* arXiv preprint arXiv:1412.6980, 2014.

Li X, Zhang W (2022) Physics-informed deep learning model in wind turbine response prediction. Renewable Energy 185:932–944

Li Z, Zheng X (2017) Review of design optimization methods for turbomachinery aerodynamics. Prog Aerosp Sci 93:1–23

Li J, Wang Y, Qiu Z, Zhang D, Xie Y (2023) Fast performance prediction and field reconstruction of gas turbine using supervised graph learning approaches. Aerosp Sci Technol 14:108425

Liu T, Li Y, Jing Q, Xie Y, Zhang D (2021) Supervised learning method for the physical field reconstruction in a nanofluid heat transfer problem. Int J Heat Mass Transf 165:120684

Lui YH, Shahriar M, Pan Y, Hu C, Hu S (2022) Surrogate modeling of acoustic field-assisted particle patterning process with physics-informed encoder–decoder approach. Struct Multidisc Optim 65(11):333

Luo J, Fu Z, Zhang Y, Fu W, Chen J (2023) Aerodynamic optimization of a transonic fan rotor by blade sweeping using adaptive Gaussian process. Aerosp Sci Technol 137:108255

Martin I, Hartwig L, Bestle D (2019) A multi-objective optimization framework for robust axial compressor airfoil design. Struct Multidisc Optim 59:1935–1947

Misaka T (2020) Image-based fluid data assimilation with deep neural network. Struct Multidisc Optim 62(2):805–814

Mohammadi-Ahmar A et al (2022) Model order reduction for film-cooled applications under probabilistic conditions: sparse reconstruction of POD in combination with Kriging. Struct Multidisc Optim 65(10):283

Novak RA (1967) Streamline curvature computing procedures for Fluid-Flow problems. J Eng Power 89(4):478–490. https://doi.org/10.1115/1.3616716

Osseyran A, Giles M (2015) Industrial applications of high-performance computing: best global practices, vol 25. CRC Press, Boca Raton

Persico G, Rebay S (2012) A penalty formulation for the throughflow modeling of turbomachinery. Comput Fluids 60:86–98

Raul V, Leifsson L (2021) Surrogate-based aerodynamic shape optimization for delaying airfoil dynamic stall using Kriging regression and infill criteria. Aerosp Sci Technol 111:106555

Salviano LO et al (2021) Sensitivity analysis and optimization of a CO 2 centrifugal compressor impeller with a vaneless diffuser. Struct Multidisc Optim 64:1607–1627

Shi D, Sun L, Xie Y (2020) Off-design performance prediction of a S-CO2 turbine based on field reconstruction using deep-learning approach. Appl Sci 10(14):4999

Sobol IM (1967) On the distribution of points in a cube and the approximate evaluation of integrals. USSR Comput Math Math Phys 7(4):86–112

Tiwari P, Stein A, Lin Y-L (2013) Dual-solution and choked flow treatment in a streamline curvature throughflow solver. J Turbomach 135(4):041004

Wagner, F., A. Kühhorn, and R. Parchem. *Robust Design Optimization Applied to a High Pressure Turbine Blade Based on Surrogate Modelling Techniques*. in *ASME Turbo Expo 2015: Turbine Technical Conference and Exposition*. 2015.

Wang Y, Liu T, Zhang D, Xie Y (2021a) Dual-convolutional neural network based aerodynamic prediction and multi-objective optimization of a compact turbine rotor. Aerosp Sci Technol 116:106869

Wang Q, Yang L, Rao Y (2021b) Establishment of a generalizable model on a small-scale dataset to predict the surface pressure distribution of gas turbine blades. Energy 214:118878

Wang Q, Zhou W, Yang L, Huang K (2022) Comparison between conventional and deep learning-based surrogate models in predicting convective heat transfer performance of U-bend channels. Energy and AI 8:100140

Wang Z, Liu X, Yu J, Wu H, Lyu H (2023) A general deep transfer learning framework for predicting the flow field of airfoils with small data. Comput Fluids 251:105738

Whitney, W.J., H.J. Schum, and F.P. Behning. *Cold-air investigation of a turbine for high-temperature-engine application. 4: Two-stage turbine performance*. 1972.

Zhao X, Gong Z, Zhang J, Yao W, Chen X (2021) A surrogate model with data augmentation and deep transfer learning for temperature field prediction of heat source layout. Struct Multidisc Optim 64(4):2287–2306