DNN-Based Surrogate Modeling-Based Feasible Performance Reliability Design Methodology for Aircraft Engine

DALU CAO AND GUANG-CHEN BAI
School of Energy and Power Engineering, Beihang University, Beijing 100191, China
Corresponding author: Guang-Chen Bai (dlxbgc@buaa.edu.cn)
This work was supported in part by the National Natural Science Foundation of China under Grant 51975028 and Grant 51776010.

ABSTRACT The risks and costs of developing a new aeroengine are fundamentally depending on the performance design final proposal. Thus, this paper presents a novel aeroengine performance design methodology that is committed to managing the effectiveness and economic availability of the design proposal. To reach such a target, the presented methodology formulates the traditional thermal cycle design problem as a reliability-based fuzzy optimization. The performance reliability is predicted by the deep neural network (DNN)-based surrogate models while a hyperparameter tuning technique is proposed to find the optimal DNN topology for better implementing a particular deep learning task. The testing results imply the DNN models with optimized topology possess remarkable function approximation capability in global so that achieves significantly higher prediction accuracy. Moreover, the DNN-based surrogate models only cost nearly 0.003% as much computing time as MC simulation (2.3591 sec vs 64746 sec, for 20 samples). Such kind of remarkably higher computational efficiency facilitates the optimization for reliability-based fitness calculation. The efficiency of the presented methodology can be further verified by abundant feasible cycle proposals. The obtained cycle solutions can achieve expected reliability (>95%) in all reference states without unnecessary performance redundancy. Besides, the diversity of feasible cycle solutions contributes to the selection of best proposal associated with engineering situation. The presented effort is favorable to acquire a more cost-efficient design proposal and enrich thermodynamic system design theory.

INDEX TERMS Aeroengine, performance reliability, cycle design, deep neural network, hyperparameter optimization, fuzzy optimization.

I. INTRODUCTION
It was inevitable: the desire for flight security always reminded the commercial aeroengine designers of the negative effects of turbomachinery degradation. Specifically, turbomachinery degradation is caused by several mechanisms including erosion, fouling and foreign objects hitting [1]. Long-term effects of these mechanisms induce uncertain changes in turbomachinery blade geometry, including blade surface damage, increased clearances, etc. These geometric changes result in the actual performance of the turbomachinery to uncertainly deviate from its designed performance, which will lead to potential overall performance loss. As an example, the potential overall performance loss would bring the following risks to the commercial aeroengine, such as insufficient net thrust and worsening fuel economy. Specifically, the former one threatens the flight performance and safety of the jetliner while the latter one increases the total fuel consumption with a drop in airline profits. Therefore, the potential overall performance loss, which is caused by degradation, remains an ongoing challenge for aeroengine.

To address this crucial issue, the practical approach is to compensate for the potential overall performance loss over the aeroengine life-cycle by maintaining overall performance redundancy. The performance redundancy comes from the further promotion of aeroengine design performance. Essentially, the modern aeroengine is a type of gas turbine engine that operates according to the Breton cycle [2]. The useful cyclic work is the source of aeroengine performance and heavily depends on two key factors. One corresponds to unit
cyclic work while the other is the mass flow of working substance. For the first factor, the unit cyclic work is strongly associated with total pressurization rate and adding heat [3]. These two items are respectively defined by some cycle parameters, including the pressurization rate of each compression component and combustor outlet temperature (or called turbine entry temperature, $T_{ET}$). Obviously, increasing these cycle parameters can effectively improve the unit cyclic work. For the second factor, there is an approximate linear growth correlation between the cyclic work and the mass flow of working substance. Such relationship points out that more air inflow can directly improve the aeroengine overall performance.

However, the reality is beyond these expectations. For instance, based on a given level of cooling technology and material type, turbine blade life is empirically halved for each 10 K rise in temperature of the incoming gas flow [4]. Undoubtedly, the higher TET cycle design scheme would bring more inconveniences for turbine blade design, such as the super-alloy selection, cooling system options and blade structure strength. Moreover, more air inflow means that the aeroengine must therefore be oversized with heavier quality. As a result, hanging the comparatively massive engines would increase the wing root bending moment and even make a possible violation in the aircraft structural strength design. Thus, the extra enhancement of cyclic work highlights the risks, since both technology backwardness and design conflict might undermine this will.

In the final analysis, the commercial aircraft engine is still a for-profit industrial product and must compete in terms of both effectiveness and economic availability. The risk of developing an entirely new aeroengine is so high that manufacturers prefer more conservative candidate proposals with mature technology. Besides, selecting the final design proposal is an extremely complex issue of cost-efficient due to the large capital expenditure and the long payback period. Thus, a variety of feasible design proposals is beneficial for the ultimate decision under an overall consideration of manufacture technique and available materials, etc.

As mentioned above, it is crucial to avoid unnecessary performance redundancy whenever possible through precisely boosting the useful cyclic work. To achieve this goal, thermodynamic cycle design needs to be formulated as an uncertainty-based design problem [5], which quantitatively considers the effect of uncertainty factors and then designs the key parameters to ensure the reliability for working conditions of interest.

Probabilistic design method is proved to be an effective method to solve the uncertainty-based design problem. Lots of relevant research is devoted to probabilistic design methods in the robustness and reliability of aerospace engineering and civil engineering [6]–[9]. These research works wisely utilized the probabilistic design method to settle the quantitative analysis of uncertainty. Then the proper key design parameters were selected to reach the reliability targets. The benefits of these studies encourage further application of probabilistic-based methodology in aeroengine performance reliability design.

Despite the above encouraging progress, a critical issue ought to be noted that hampers the further application of probabilistic design method. To be specific, the key to implementing probabilistic-based analysis and design is to acquire the probability distributions of the concerned parameters. In the pursuit of this target, the essential process is predefined the uncertainty factors and then performing the randomized trials based on Monte Carol (MC) simulation [10]. However, MC simulation is an extremely expensive computational process that requires a huge investment in time. The disadvantageous effect of this characteristic is further amplified in the optimization problem, which requires calculating the reliability-based fitness by MC simulation for all particles (or individuals) in each iteration (or generation). Thus, it would cost unaffordable time to carry out the reliability-based optimization design if hundreds or even thousands of MC simulations are needed.

In order to avoid the time-consuming MC simulation, considerable attention has been currently paid to the surrogate model technique. The surrogate model is explained as a black box program which only imitates the mapping relationship of the real physical models based on some particular mathematical models, such as polynomials [11], support vector machine [12] and Machine Learning (ML) [13]–[15], etc. One branch of ML is deep learning, which has lately attracted extensive interest. Deep learning is usually implemented using a deep neural network (DNN) architecture. The deep is interpreted as the number of layers in the network, which endows the DNN with impressive capability for learning representations of data with multiple levels of abstraction [16]. This remarkable ability makes the DNN become the ideal candidate for the surrogate models.

In this study, the DNN is utilized to develop surrogate models for predicting aeroengine performance reliability in each concerned state. Moreover, to further release the potentials of DNN, MINLP-based hyperparameter tuning technique is proposed to search for the optimal DNN topological structure. Employing the DNN-based surrogate models with optimum topology, the reliability-based fuzzy optimization problem is formulated to solve the feasible thermodynamic cycle proposals. Then the proposed approach is applied to solve the commercial aeroengine conceptual design problem with uncertainty effects.

This paper is structured as follows: Section 2 illustrates the modeling technique for the commercial aeroengine. In Section 3, implementation of MINLP-based hyperparameter tuning technique is introduced in detail with its application in DNN training procedure. Section 4 investigates the utilization of DNN-based surrogate models in order to formulate fuzzy optimization for commercial aeroengine performance reliability design. Section 5 quantitatively discusses the optimization results to validate the proposed methodology. Conclusions and perspectives are given in Section 6.
II. COMMERCIAL AEROENGINE MODELING

The aim of aeroengine modeling is to investigate the probability distributions of overall performance parameters associated with the specified cycle parameters and predefined uncertainty. It is necessary to yield the training sets, which are the combinations of thermal cycle and reliability output responses.

A. OVERVIEW OF COMMERCIAL AEROENGINE

This work studies the high bypass ratio dual-spool turbofan engine, which is an extensively-used type to power commercial jetliners. A simplified cross-section of this engine is displayed in Fig. 1, with its major components marked.

The operation mechanism and layout could be briefly described as follows. When the airflow accesses the engine inlet, it is divided into bypass stream and core stream. The bypass stream is compressed and then directly exhausted to produce the main thrust by increasing the inflow momentum through the engine. Besides, a small amount of thrust is generated by the core stream, which is also employed as the working medium for multistage compression, heating, and expansion processes. The compression and expansion are split into parts by LP spool and HP spool. Mounted on the LP spool, fan and booster stage are driven by LP turbine. Connecting through the HP spool, the HP turbine supplies power to actuate the HP compressor. The kinetic energy source for turbo machines operating comes from burning the fuel in the combustor.

B. MODELING PRINCIPLE

Based on the principles of aerodynamics and thermodynamics, this study developed the component-level steady-state performance simulation model. This type of model is widely-used in the conceptual design phase of aviation propulsion system. Moreover, this model is constructed as a modular architecture to improve the modeling efficiency, as shown in Fig. 2.

For evaluating the uncertainty performance behavior in concerned conditions, the established model is composed of two principal parts, including uncertainty turbomachinery performance module and turbofan engine prototype model.

C. UNCERTAINTY TURBOMACHINERY PERFORMANCE MODULE

This module is embedded into the turbofan engine prototype model to simulate the uncertainty turbomachinery performance owing to degradation. The effects of above uncertain changes are represented by changes in turbomachinery...
TABLE 1. Range specifications of uncertainty turbomachinery performance parameters.

| Turbomachinery component | Noise parameter(σ) | Lower | Nominal | Upper |
|--------------------------|--------------------|-------|---------|-------|
| Fan, Booster, Compressor, HP turbine, LP turbine | Efficiency | -3σ | 1 | -3σ |
|                          | Mass flow capacity | -3σ | 1 | -3σ |

performance parameters, including isentropic efficiency ($\eta_i$) and mass flow capacity ($W_{cor}$). To achieve above uncertain changes of parameters, the relevant measure is to cause random numerical oscillation of theoretical design performance. The theoretical design performance is depicted by the standard maps. These maps represent the technical characteristics of a real turbo machine on the test bench [17]. With the instance of fan, Fig. 3 displays the possible change ranges of efficiency and mass flow capacity based on its standard map.

According to the references [18], [19], these uncertainty parameters could be numerically described by the stochastic uncertainty and obey the normal distribution. Thus, the numerical model of efficiency and mass flow capacity are

$$[\eta_i, W_{cor}] = \begin{bmatrix} r_1 S \eta_{i, Map} \\ r_2 S W_{cor, Map} \end{bmatrix}, \quad r \sim N(\mu, \sigma)$$

where, $S$ is the scaling coefficient later in Section 2.4.1; $r$ is the random number predefined to obey normal distribution.

Due to the absence of turbomachinery monitoring data from airline company, this investigation bounds the numerical range of uncertainty performance parameters based on the 3-σ principle [20], as shown in Table 1.

**D. TURBOFAN ENGINE PROTOTYPE MODEL**

As the principle part of the developed simulation model, the prototype model is jointly organized by in-design point simulation and off-design point simulation.

1) IN-DESIGN POINT SIMULATION

The in-design point simulation investigates a certain configuration aeroengine of many possible thermal cycle schemes. Multiple thermodynamic cycle alternatives are studied to comparatively evaluate the overall performances in one reference operating state. Once a thermal cycle candidate is elected, the technical parameters of each turbomachinery component are designed. To be specific, the turbomachinery standard maps are scaled to associate the designed component performances in the design point. In addition, the chosen cycle also decides the feature sizes of aeroengine, such as the throat area of exhaust. In summary, in-design point simulation practically designs the geometry of the aircraft engine, which is prior to the off-design point simulation.

2) OFF-DESIGN POINT SIMULATION

Off-design point simulation calculates the performance of the designed aeroengine under the numerous operating states that described by the regulation schedule, altitude and Mach number in flight. In a particular operating state, the critical issue for off-design point simulation is to determine the collaborative operating condition.

For the studied commercial aeroengine, its collaborative operating condition needs seven decision elements to determine whose vector format could be given as

$$\Psi = [\psi_1, \psi_2, \psi_3, \psi_4, \psi_5, \psi_6, \psi_7]^T$$

where, $\psi_i (i = 1, \ldots, 5)$ are the auxiliary coordinates of each turbomachinery component in its scaled maps, which depict the locations of operating point representing the in-service condition; $\psi_i (i = 6, 7)$ are two of the $N_{HP}$, $N_{LP}$ and $N_t$, which are chosen according to the user-defined regulation schedule.

To ensure that there is only one solution of decision elements to match one unique operating state, the collaborative operating condition is represented by seven nonlinear implicit equations. These equations supervise the mass flow compatibility among all parts and the power balance for each spool. According to the sections in Figure 2, the equations are formulated as the residual function in the programming and presented as

(1) The mass flow residual function of internal duct of fan and booster

$$want f_1 (\Psi) = (W_{21} - W_{24})/W_{24} \to 0$$
The mass flow residual function of booster and compressor
\[ \text{want } f_2(\psi) = \frac{(W_{24} - W_3 - W_{cool})}{W_{24}} \rightarrow 0 \]  

The mass flow residual function of the compressor and HP turbine
\[ \text{want } f_3(\psi) = \frac{(W_5 - W_3 - W_f - W_{cool,LP})}{W_5} \rightarrow 0 \]  

The mass flow residual function of bypass and external duct of nozzle
\[ \text{want } f_4(\psi) = \frac{(W_8 - W_6)}{W_8} \rightarrow 0 \]  

The mass flow residual function of LP turbine and internal duct of nozzle
\[ \text{want } f_5(\psi) = \frac{(W_{16} - W_{18})}{W_{16}} \rightarrow 0 \]  

The power residual function of the HP spool
\[ \text{want } f_6(\psi) = \frac{(\eta_{HP}P_{HPT} - P_{Compr})}{\eta_{HP}P_{HPT}} \rightarrow 0 \]  

The power residual function of the LP spool
\[ \text{want } f_7(\psi) = \frac{(\eta_{LP}P_{LPT} - P_{Fan} - P_{Booster})}{\eta_{LP}P_{LPT}} \rightarrow 0 \]  

A nonlinear implicit equation group is composed of all the residual equations and formulated as
\[ F(\psi) = [f_1(\psi), \ldots, f_7(\psi)]^T \]  

Thus, solving this equation group is the key to determining the collaborative operating condition, and is presented as
\[ \text{solved } \psi \text{ for } \| F(\psi) \|_\infty \leq \varepsilon, \quad \forall \varepsilon > 0 \]  

where, \( \varepsilon \) is precision factor and its value is set as \( 10^{-5} \).

Then the Newton-Raphson algorithm is adopted to solve the \( \psi \), which is updated in each iteration [21].
\[ \psi^{(k+1)} = \psi^{(k)} - (J^{(k)})^{-1} F(\psi^{(k)}) \]  

where, \( J \) is the Jacobi matrix that contains first derivatives of the residual errors with respect to the decision elements. Moreover, \( J \) is formulated as
\[ J = \begin{bmatrix} \frac{\partial f_1}{\partial \psi_1} & \cdots & \frac{\partial f_1}{\partial \psi_7} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_7}{\partial \psi_1} & \cdots & \frac{\partial f_7}{\partial \psi_7} \end{bmatrix} \]

More details of modeling techniques of each component can be found in these references [22]–[24].

III. DNN-BASED SURROGATE MODEL FOR RELIABILITY PREDICTION

In the reliability-based optimization, it would cause the expensive time cost to calculate each individual fitness by performing the MC simulation. To replace the time-consuming MC simulation, DNN is exploited to construct surrogate models for predicting the aeroengine performance reliability under multiple flight conditions. Meanwhile, MINLP-based hyperparameter tuning technique is presented to determine the optimal DNN topological structure for enhancing network performance.

A. DEEP NEURAL NETWORK

By abstracting the mechanism of human brain functioning, Artificial neural network (ANN) construct artificial neurons and establish the connections between artificial neurons according to a certain topology structure. Through network topology structure, ANN is intended for the approximation of the implicit non-linear relationship between input and output variables through extracting the features of the training scenarios.

The ANN formulation between output response \( \hat{y} \) and input variables \( x \) can be structured as
\[ \text{given: } D_{\text{train}} = \left\{ \left( x^{(1)}, y^{(1)} \right), \ldots, \left( x^{(N)}, y^{(N)} \right) \right\} \]  
\[ \text{want: } f_{\text{ANN}}(x, D_{\text{train}}, \Theta) = \hat{y} \approx y \]  

where, \( f_{\text{ANN}}(\cdot) \) is the ANN model; \( D_{\text{train}} \) is a training set; \( \Theta \) is the vector of hyperparameters that will be further explored in the following text.

Fig. 4 displays the topology structure of shallow neural network, which is a classical type of ANN. It is composed of an input layer, a hidden layer as well as an output layer. Each layer is connected to the front and back layers by neurons, using the output of the front layer as its input.

DNN is a special form of ANN. Compared with the shallow neural networks, the most significant characteristic of DNN is that it contains at least two or even hundreds hidden layers, as shown in Fig. 5. In this deep-layered, hierarchical structure, multiple nonlinear transformations are implemented to fully extract the features of training data set [25]. The more accurate presentation of data characterizations makes the DNN more capable at expressing the highly complex nonlinear relationship between input and output values.

Despite the remarkable advantage mentioned above, the unprecedented complexity of the DNN topology structure makes it more difficult for hyperparameters tuning. The definition of hyperparameter is the predefined network parameters that programmers ought to specify before ANN training [26]. These presetting design the topology structure, and have profoundly influences on training converge and generalization ability of a neural network [27]. Therefore, the hyperparameters must be appropriately adjusted for training a high-precision DNN.

At present, the leading approach for hyperparameters searching is still the manually adjusting referred to the prior
knowledge from previous tuning observations [28]. Frankly speaking, this is already an extremely time-consuming and painful work for even an experienced programmer to obtain a well-tuned shallow neural network. However, a well-tuned DNN is almost impossible to acquire by manual adjustment due to the increase of the number of hidden layers. Taking the 6-hidden-layers DNN as an example, there are 64,000,000 (=10^6 × 2^6) practicable combinations to design the topological structure, if the \(N_{hl}\) ranged from 2 to 11 and chosen one of the two activation functions for each hidden layer. Hence the previous manual hyperparameters tuning method is facing challenge and it is worthwhile devoting much effort to this.

\section*{B. MINLP-BASED HYPERPARAMETER TUNING TECHNIQUE}

In order to acquire a well-tuned DNN automatically and efficiently, a new hyperparameter tuning technique is proposed based on mixed integer nonlinear program using Mesh Adaptive Direct Search algorithm.

\subsection*{1) HYPERPARAMETERS OF DEEP NEURAL NETWORK}

Three kinds of key hyperparameters, including network topological parameters, learning rate and activation function of each layer are focused in the presented technique as follows.

\subsection*{a: NETWORK TOPOLOGICAL PARAMETERS}

The feedforward DNN is investigated and its structure is graphically described in Fig. 4. The structure of DNN is defined by network topological parameters, including the number of layers, the number of neurons in each layer.

\subsection*{b: LEARNING RATE}

Based on the gradient descent algorithm, learning rate determines the size of search step for updating connection weights in network training process. If the learning rate is too small, the network training process will converge very slowly. On the contrary, it will cause oscillations near the optimum convergence point, which is difficult to eliminate or even results in training failure. Empirically, its value setting ranges from 0.0001 to 1 that depends on network topology and training data.

\subsection*{c: ACTIVATION FUNCTION}

The activation function performs the nonlinear transformation that represents the mapping relationship between input and output response of neurons. Thus, the selection of activation function in output layer and each hidden layer deeply affects the network performance of training and generalization. Three types of classical activation functions, including sigmoid, tanh and purelin [29], are of our major concern. The images and formulations of the mentioned activation functions are displayed in Fig. 6.

\subsection*{2) PROBLEM DEFINITION OF HYPERPARAMETER TUNING}

In essence, the aim of hyperparameter tuning is to find a proper hyperparameter combination at a finite admissible set. Therefore, the hyperparameter tuning procedure could be defined as a combinatorial optimization problem. Its optimization objective can be set to minimize the training error of loss function by searching for an optimal hyperparameter solution at a finite admissible set. Therefore, the optimization formulation for hyperparameter tuning can be presented as

\[
\text{minimize } \sum_{i=1}^{l} L\left(y^{(i)}_{\text{test}}, f\left(x^{(i)}_{\text{test}}, D_{\text{train}}, \Theta\right)\right)
\]

s.t. : \(\theta_j \in [\theta_{jl}, \theta_{ju}], \quad j = 1,\ldots, N_{hl} + 1;\)

\(\theta_j \in \{p \in \mathbb{Z} | Z_l \leq p \leq Z_u\}, \quad j = 2,\ldots, N_{hl} + 1;\)

\(\theta_j \in \{\text{sigmiod}, \text{tanh}, \text{purelin}\}, \quad j = N_{hl} + 2,\ldots, 2N_{hl} + 2.\) \quad (15)

where, \(L(\cdot)\) is the loss function that measures the fitting ability; \(Z\) represents the positive integer number set.

This optimization problem is illustrated as follows. The DNN is a highly nonlinear model and its loss function is a non-convex function, so loss minimization is a non-convex nonlinear optimization problem. Obviously, the learning rate is a continuous variable bounded in a specified numerical range. However, it is so noticed that \(N_{hl}\) of each hidden layer must be set to positive integer. Moreover, the activation function selection of each layer is a categorical variable for which the measurement scale is a set of activation function categories.

Above discussions indicate that hyperparameter optimization ought to be formulated as a Mixed Integer Nonlinear Optimization problem (MINLP) of network hyperparameters. Next, we introduce a new hyperparameter tuning technique based on MINLP formulation.
Program (MINLP), which contains the nonlinear objective function and combines continuous variables with categorical variables or even ordinary discrete variables such as positive integer. Owing to different types of decision variables, the MINLP for hyperparameter tuning fails to be solved within the some frequently-used optimizers, such as gradient descent method and various types of evolutionary algorithms.

3) MESH ADAPTIVE DIRECT SEARCH ALGORITHM

To address the above problem, the Mesh Adaptive Direct Search algorithm (MADS) is exploited as the underlying algorithm to perform the MINLP-based hyperparameter tuning [30], [31].

The implementation of MADS algorithm basically depends on the generation of mesh, which realizes the discretization of the design space. The discrete structure of mesh at $k$th iteration is defined as

$$M_k = \bigcup_{x \in V_k} \{ x + \Delta^m_k D z : z \in N^{nD} \}$$

(16)

where, $V_k$ is the set of points where the objective function and constraints have been evaluated by the start of iteration $k$; $\Delta^m_k$ is the size parameter of mesh; $z$ is a $n$-dimensional integer vector; $D$ is the set of mesh directions and is constructed as a $n \times n_{DO}$ matrix representing a fixed finite set of $n_{DO}$ directions, an integer vector.

In a series of varying-size meshes, MADS algorithm implements an adaptive seek iteratively. Each iteration contains two steps, including the search and the poll. Through the user-defined search strategy, such as LHS [32] and VNS [33], the search step generates multiple testing points anywhere on the mesh to find a better solution. If the current search step fails to achieve this goal, the poll step is performed to explore the mesh near the $x_k$. The following set of poll trial points is strictly defined as

$$P_k = \{ x + \Delta^p_d d : d \in D_k \} \subset M_k$$

(17)

where, $D_k$ is the set of poll directions; $\Delta^p_k$ is the poll size parameter;

To be specific, the poll trial points are generated at the place that is in the vicinity of the best current solution $x_k$. The distances between $x_k$ and poll trial points are limited by the size parameter of poll. Within two-dimension mesh, Fig. 7 displays the locations of poll trial points in three iterations and graphically compares the size parameters of the mesh and poll. For the next iteration, the mesh size parameter is also updated with

$$\Delta^m_{k+1} = \kappa \omega_k \Delta^m_k$$

(18)

where, $\kappa < 1$ is a fixed positive. $\omega_k$ is strictly negative if the iteration $k$ failed to find a better solution, or positive if iteration $k$ success. In other words, when an iteration fails to improve the current best solution by search or poll, the next iteration is initiated on a finer mesh. On the contrary, a coarser mesh is desirable.

In Table 2, the implementation of MADS is briefly illustrated within the pseudo code.

| TABLE 2. Pseudo code of MADS. |
|--------------------------------|
| initialize $x_0 \in V_0$ set $\Delta_{x0}^m$, $\Delta_{x0}^p$, $x_{sept} = x_0$ and $k = 0$
| while stopping criteria is not satisfied
| search on the mesh to find a better solution than $x_k$
| if search failed then
| poll on the mesh to find a better solution than $x_k$
| end
| if the expected success is achieved by either the search or the poll then
| set $x_{sept} = x_{x+1}$, $w_k < 0$, $\Delta_{x+1}^m = \kappa \Delta_{x}^m$
| else
| set $x_{sept} = x_k$, $w_k > 0$, $\Delta_{x+1}^m = \kappa \Delta_{x}^m$
| end
| update $\Delta_{x+1}^m$ and $\Delta_{x+1}^p$, set $k = k + 1$
| end

C. DNN TRAINING PROCEDURE

Based on the LM back-propagation algorithm [13], network training iteratively adjusts the network parameters (mainly connecting weights) to approximate the mapping relationship of input and output response from the feeding training sets. To acquire qualified training sets, this subsection investigates the necessary mathematical skills to preprocess the original data, including feature scaling and statistical analysis.
Nevertheless, this subsection also studies the overview of DNN training with the application of MINLP-based hyperparameter tuning.

1) FEATURE SCALING FOR INPUT OF TRAINING SET
The numerical range differences of input elements cause the gradient direction more easily deviating from the optimal search direction, which is unfavorable to the efficiency of network training. To address this issue, feature scaling is exploited to eliminate this effect of scale by normalizing the dimension of each decision variable to the same numerical range [34]. Using the Z-score normalization, the normalized value of samples \( x(i), i = 1, \ldots, n \) in the \( j \)th dimension is

\[
\hat{x}_j(i) = \frac{x_j(i) - \bar{x}_j}{\sigma_j}
\]

where, \( \bar{x}_j \) is the mean; \( \sigma_j \) is the standard deviation.

Through the above numerical transformation, each dimension of \( x(i) \) is centered to mean 0 and scaled to standard deviation 1.

2) STATISTICAL ANALYSIS FOR OUTPUT OF TRAINING SET
Based on the turbofan model mentioned in Section 2, the MC simulation is undertaken to acquire the data of the concerned performance parameters. Referring to the obtained data with specified performance benchmarks, statistical analysis is performed to calculate the performance reliability of the predefined key design parameters.

The reliability value \( R \) of the \( i \)th scenario in the \( k \)th state is presented as

\[
R^{(i)}_k = \Phi \left( \frac{\xi^{(i)}_k}{\sqrt{\Gamma^{(i)}_k}} \right)
\]

where, \( \Phi(\cdot) \) is the standard normal distribution function; \( \xi \) and \( \Gamma \) are the functions to calculate mean and standard deviation, and respectively formulated as

\[
\xi_k^{(i)} = \frac{1}{N} \sum_{i=1}^{N} \left( y_k(x^{(i)}, r) - y^*_k \right)^2
\]

where, \( r \) is the random series characterized by a defined probability distribution; \( y(\cdot) \) is the performance output response of \( k \) state; \( y^* \) is the performance benchmark.

To sum up, the preprocessed training set \( D_{\text{train}} \), which associates normalized input with the corresponding reliability outputs in all states, is

\[
D_{\text{train}} = \{ (\hat{x}^{(1)}, R^{(1)}), \ldots, (\hat{x}^{(N)}, R^{(N)}) \}
\]

3) OVERVIEW OF ESSENTIAL PROCESS
As shown in Fig. 8, the DNN training procedure integrates three principal phases steps, including the training sets pre-processing, network training and MINLP-based hyperparameter optimization.
The first stage is to preprocess the data of MC using feature scaling and statistical analysis. In the second stage, MIMLP-based hyperparameter tuning initializes or updates the hyperparameters to refresh the topology of DNN. As for the third stage, LM back-propagation algorithm is adopted to update the network parameters (connection weights and threshold value) [35]. Both the second stage and third stage will iteratively carry out until meet the user-defined stopping criteria, including the error convergence or reach the maximum number of iterations.

**IV. SURROGATE MODELING-BASED FUZZY OPTIMIZATION**

Based on the already trained DNN-based surrogate models, this section studies the implementation of fuzzy optimization. The aim of optimization is to acquire the multiple feasible cycle solutions of the commercial aeroengine.

**A. FLIGHT MISSION ANALYSIS**

Fig. 9 displays the typical commercial flight mission profile. It graphically describes the basic capabilities of a jetliner by performing the following multiple specific missions. Some critical missions are conducive to explore the leading performance requirements of installed engine.

First of all, it is the top priority for flight security to assure that the aircraft could accelerate to take-off speed in the limited distance. To achieve this goal, the designed aeroengine is requested to produce sufficient thrust in airplane ground rolling, even under some adverse weather conditions. Secondly, it spends the majority of flight time in the cruise phase for the passage plane. Thus, the aeroengine thrust in cruise occupy an essential position of the flight performance and safety. Thirdly, the specific fuel consumption in cruise is directly related to the economy of the entire jetliner flight. Fourthly, flight at the top of climb is approaching the performance envelope of the equipped engine so that it is necessary to be considered. However, there is no need to overemphasize the flight at the top of climb because it is not always an indispensable flight mission. In the end, the rest of specific missions usually are transient states that are completed in a short time.

As mentioned above, this study mainly concentrates on three operating conditions, including take off, cruise and top of climb. Table 3 lists the specifications of the concerned operating conditions, which are depicted by flight condition, regulation schedule and corresponding overall performance requirements. The International Standard Atmosphere (ISA) defines that $T_{\text{amb}} = 288.15K$, $P_{\text{amb}} = 101.3kPa$, which is the setting of the standard sea-level atmospheric condition.

**B. OPTIMIZATION PROBLEM STATEMENT**

The implementation of optimization is to solve the feasible cycle solutions for reaching the requested overall performance requirements under the consideration of uncertainty.

1) **OBJECTIVE**

The primary purpose of optimization is to maximize the overall performance reliability of each operating condition as much as possible. The objective functions are presented as

$$\max : \left\{ \frac{f_j(x)}{f_{\text{DNN}}(x)} \right\} \quad j = 1, 2, \ldots, m \quad (24)$$

2) **DECISION VARIABLES**

Thermodynamic cycle defines the overall performance at all operating conditions, which is one of the most important attributes for a gas turbine engine [36]. Therefore, the decision variables are the cycle parameters of on-design point (in SLS condition), including the pressurization rate of fan ($\pi_F$), booster ($\pi_B$) and compressor ($\pi_C$), turbine entry temperature ($T_{\text{ET}}$), bypass rate ($R_{\text{BP}}$) and air inflow of inlet ($W_{\text{in}}$). The vector format of decision variables could be presented as

$$x = [\pi_F, \pi_B, \pi_C, T_{\text{ET}}, R_{\text{BP}}, W_{\text{in}}]^T \quad (25)$$
TABLE 3. Overall performance requirements in flight mission profile.

| Operating condition | Atmospheric environment | Alt (ft) | Ma | Regulation schedule | Overall performance requirements |
|---------------------|-------------------------|---------|----|---------------------|----------------------------------|
| SLS                 | ISA                     | 0       | 0  |                     |                                  |
| Tko                 | Hot day, +15K           | 35      | 0.15 | Fixed Ni            | ≥ FRM, Tko                       |
| Cru                 | ISA                     | 35000   | 0.78 | Fixed Ni            | ≥ FRM, Cru                       |
| ToC                 | ISA                     | 41000   | 0.80 | Fixed Ni            | ≥ FRM, ToC                       |

TABLE 4. Cycle parameter design range.

| Cycle parameter (x) | π_F (x_i) | π_s (x_i) | π_C (x_i) | T_{ET} (x_i) | R_{BP} (x_i) | W_{o} (kg) |
|---------------------|-----------|-----------|-----------|-------------|-------------|------------|
| Origin cycle        | 1.60      | 2.21      | 10        | 1600        | 5.1         | 353.2      |
| Lower               | 1.5       | 1.8       | 9         | 1550        | 4           | 330        |
| Upper               | 2.0       | 2.5       | 12        | 1700        | 6           | 380        |

1) NORMATIVE FORMULATION OF FUZZY OPTIMIZATION

The normative fuzzy optimization model is composed of three principle elements: A finite set of feasible alternatives; a set of fuzzy constraints which softly restrain the selection of alternatives; the fuzzy objective functions that quantify the satisfaction of each alternative. Referred to the above presentation, the normative formulation of this study is presented as:

\[
\text{find : } x = [x_i] \\
\text{max : } \left\{ f^{(l)}_{DNN}(x) \right\} \\
\text{s.t. : } \left\{ \begin{array}{l}
            x_l \leq x_i \leq x_u \\
            R_W(x) \leq 0.9 
\end{array} \right. 
\]

where, \( f_{DNN} \) is the DNN-based surrogate model; subscript \( l \) and \( u \) represent lower and upper limits for design variables, respectively.

2) IMPROVED Z-SHAPED MEMBERSHIP FUNCTION

In fuzzy programming, the feasible solutions are solved with imprecise objectives and constraints, which are not measured by specific value but in terms of satisfaction (or called desirability). The degree of satisfaction needs to be quantified at the level of mathematical scale and then transferred into an equivalent real-valued model. This process is called defuzzification. One practicable defuzzification method is associating satisfaction with membership function. Thus, the membership function quantifies corresponding satisfactory of objectives and constraints. Based on Z-shaped membership function [40], the improved Z-shaped membership function is defined by the following formulas:

\[
\mu(\tau, \chi) = \begin{cases} 
1 & \tau \leq \tau_l \\
1 - \left( \frac{\tau - \tau_l}{\tau_u - \tau_l} \right)^q & \tau_l \leq \tau \leq \tau_u \\
0 & \tau \geq \tau_u 
\end{cases} 
\]

where, \( \mu \) is the defuzzified satisfaction degree. \( \tau \) is the objective value; \( \tau_u \) is the completely satisfactory objective value while \( \tau_l \) is the completely unsatisfactory objective value.

Fig. 10 presents the images of Z-shaped membership function and improved Z-shaped membership function. Obviously, the notable feature of improved Z-shaped membership
function that is introducing q to reshape the partially satisfied region.

The value setting of q is regarded as the weight to quantify the importance of each objective. If an objective has a greater significance, the value of q will be set larger. Otherwise, q will be equal to a relatively smaller value.

3) REAL-VALUED FORMULATION OF FUZZY OPTIMIZATION PROBLEM

With different parameter settings, the presented membership function is exploited to defuzzify the desirability of objectives and constraints. Furthermore, the defuzzified constraint is transformed as a pseudo objective function. This transformation assists the programmer to weigh the feasibility and desirability of a solution. Therefore, these two objective functions make the studied problem become a multi-objective problem, which can be reformulated as

$$\text{find} : x = \{x_i\}$$
$$\min : \{\lambda, \beta\}$$
$$s.t. : \begin{cases} x_{il} \leq x_i \leq x_{iu} \\ \lambda = \sum_{j=1}^{4} \mu_j(f_{DNN}^{(j)}(x), q^{(j)}) \\ \beta = \mu(R_W(x), q_g) \end{cases}$$

where, $\lambda$ is the first objective value that represents the satisfactory value of each objective; $\beta$ is the second objective value that represents the satisfactory value of each constraint.

According to the discussion of section 4.1, the value setting of $q^{(j)}$ refers to the order of importance, which is,

$$q_{R_{Fn,ToC}} < q_{R_{SFC,Cru}} < q_{R_{Fn,Cru}} = q_{R_{Fn,Tho}} < 1$$

For comparison, the studied problem is also formulated as a mirrored deterministic optimization. Its objective function is developed by integrating the reliability prediction responses of DNN models. Therefore, the deterministic optimization is formulated as

$$\text{find} : x = \{x_i\}$$
$$\min : \{\lambda, \beta\}$$
$$s.t. : \begin{cases} x_{il} \leq x_i \leq x_{iu} \\ \lambda = \sum_{j=1}^{4} w_j \left| 1 - f_{DNN}^{(j)}(x) \right| \\ \beta = R_W(x) \end{cases}$$

where, $w$ is the weight of each objective and has a similar value setting with $q$.

V. RESULT AND DISCUSSION

A. VALIDATION OF COMMERCIAL AEROENGINE MODEL

Referring to the performance characteristics of CFM56-7B [41], the overall performances of concerned states are calculated by both the established turbofan model and Gasturb® [42], as a comparison. The comparative results are listed in Table 5.

The results imply that the errors between both models are within the acceptable range of engineering calculation. Thus, it verifies the simulation accuracy of the developed aeroengine model. Besides, the results of the established model are used as the performance benchmarks of each working condition.

B. PERFORMANCE ASSESSMENT OF DNN-BASED SURROGATE MODELS

In this section, the prediction accuracy of the obtained DNN-based surrogate models is evaluated to validate the effectiveness of MINLP-based hyperparameter tuning technique. In order to effectively cover the entire design space, the number of training scenarios shall be 30 times the dimension of the input [3]. Thus, this work generates 180 training scenarios and another 20 testing scenarios. Feeding these samples, DNN-based surrogate models are training with the

**TABLE 5. Simulation comparison results of Gasturb® and developed turbofan model.**

| Operating condition | performance parameter | Gasturb® | This work | Error (%) |
|---------------------|-----------------------|---------|-----------|-----------|
| Tko                 | $P_f$ (kN)            | 110.44  | 116.52    | 5.51      |
| Cru                 | $P_f$ (kN)            | 22.12   | 23.25     | 5.11      |
| SFC                 | $SFC$ (kg/h*Kn)       | 62.92   | 61.57     | 2.14      |
| ToC                 | $P_f$ (kN)            | 16.46   | 16.98     | 3.16      |
employment of MINLP-based hyperparameter optimization method. The fitness curves of hyperparameter optimization are presented in Fig. 11.

Selecting the best hyperparameter scheme of $R_{Fn,Tko}$ is presented for the sake of illustration. In this case, MADS approach the optimal hyperparameter combinations, which are specifying the $N_{HL}$ ranged from 3 to 6. Then, the network performances of each combination are evaluated through 20 testing samples, which are measured by the 2-norm and infinite-norm of prediction errors. The following results, including the optimum hyperparameter schemes and accuracy observations, are shown in Table 6.

Obviously, the best accuracy is observed from the optimal hyperparameter combination of $N_{HL} = 3$. Thus, this hyperparameter scheme is selected to design the topological structure of $R_{Fn,Tko}$ DNN-based surrogate model. Likewise, the optimum topology of the rest DNN-based surrogate models are also determined, as shown in Table 7.

Surveying the selected hyperparameter schemes for each DNN model, it is found that they are very different in all

---

**FIGURE 11.** Fitness curves of MINLP-based hyperparameter optimization.

**TABLE 6.** Hyperparameters schemes and accuracy observations of $R_{Fn,Tko}$.

| $N_{HL}$ | LR  | Hidden layers | Output layer | Accuracy |
|---------|-----|---------------|--------------|----------|
|         |     | $N_{lay}$ | $\sigma_1$ | $\sigma_2$ | $\sigma_3$ | $\sigma_4$ | $\sigma_5$ | $\sigma_6$ | $\|err\|$ | $\|err\|$ |
| 3       | 0.111 | 12 | I | 3 | I | 10 | II | \ | \ | \ | \ | \ | I | 0.0192 | 0.0155 |
| 4       | 0.598 | 9  | II | 9 | II | 7 | 1 | 4 | I | \ | \ | \ | \ | I | 0.0219 | 0.0173 |
| 5       | 0.212 | 7  | I | 5 | II | 5 | 1 | 6 | I | 5 | II | \ | \ | I | 0.0399 | 0.0197 |
| 6       | 0.102 | 12 | II | 6 | I | 6 | 1 | 10 | II | 9 | II | 6 | I | III | 0.0427 | 0.0218 |

$\sigma_1$: Activation function selection. I: sigmoid. II: tanh. III: purelin.
This diversity indicates that different deep learning tasks always require different hyperparameter combination to achieve the best effects.

A further test is implemented to compare the network performances of the well-tuned DNN models and the non-tuned DNN models with the same $N_{HL}$. For the non-tuned DNN models, the default topological structure is that each hidden layer has six neurons and uses the sigmoid, as well as its output layer exploits the purelin. The results of this comparison are graphically illustrated in Figure 12. For better demonstration effect, the testing samples are orderly arranged according to their reliability values from 0 to 1.

As shown in Fig. 12, the non-tuned DNN models reach basically acceptable accuracy to predict some testing samples whose target value is equal to 0 or 1. However, there are significant inaccuracies when the non-tuned DNN models predict the testing samples whose target value range from 0.1 to 0.9. Different from the non-tuned DNN models, the predictions of the tuned DNN models are highly consistent with the target values of all testing samples. It indicates the well-tuned DNN models have accessed to excellent function approximation capability in global. While the non-tuned DNN models are missing this ability, and only barely qualify for pattern classification if all the samples are labeled 0 or 1.

In this work, all the simulating calculations are performed by one computer equipped with an Intel CPU (E5 2680v2) and 16-GB RAM. Based on this machine, the computing times are measured to compare the computational efficiency of MC simulation and DNN-based surrogate models. The Table 8 demonstrates the total time cost for evaluating 20 testing samples, as well as the accuracy of DNN-based surrogate models.

The results show that the maximal absolute error is not beyond 0.042. Thus, the DNN-based surrogate models reach acceptable accuracy. In looking at the specifications for the computing time, it is worth noting that MC simulation takes 64746 sec to complete the comprehensive reliability prediction for 20 cycle samples. However, the DNN-based surrogate models only spend 2.359 sec to finish the same task. Therefore, the DNN-based surrogate models are capable of replacing the MC simulations for probabilistic analysis, with satisfied prediction precision and significant computational efficiency improvement.
TABLE 7. Optimum hyperparameter schemes for each deep learning task.

| DNN-based surrogate model | 1st | 2nd | 3rd | 4th | 5th | 6th | Output layer | \(\sigma(\cdot)\) |
|---------------------------|-----|-----|-----|-----|-----|-----|-------------|---------------|
| \(R_{\text{mc,mc}}\)     | 0.111 | 12 | 3 | 1 | 10 | \(\Pi\) \(\Pi\) \(\Pi\) \(\Pi\) \(\Pi\) \(\Pi\) \(\Pi\) \(\Pi\) \(\Pi\) | I |
| \(R_{\text{mc,ToC}}\)    | 0.102 | 8 | \(\Pi\) | 7 | \(\Pi\) | 6 | \(\Pi\) | 5 | 1 | 8 | \(\Pi\) \(\Pi\) \(\Pi\) | I |
| \(R_{\text{ToC,mc}}\)    | 0.594 | 12 | \(\Pi\) | 11 | \(\Pi\) | 5 | \(\Pi\) | 6 | \(\Pi\) | 11 | \(\Pi\) | 11 | \(\Pi\) | \(\Pi\) | \(\Pi\) | III |
| \(R_{\text{mc,Cru}}\)    | 0.109 | 11 | \(\Pi\) | 3 | \(\Pi\) | 5 | \(\Pi\) | 6 | \(\Pi\) | 2 | 1 | 11 | \(\Pi\) | 6 | \(\Pi\) | I |

\(\sigma(\cdot)\): Activate function selection. I: \textit{sigmoid}. \(\Pi\): \textit{tanh}. III: \textit{purelin}.

TABLE 8. Assessment of accuracy and computational efficiency.

| \(R_{\text{mc,mc}}\) | \(R_{\text{mc,ToC}}\) | \(R_{\text{ToC,mc}}\) | \(R_{\text{mc,Cru}}\) | Computing Time (s) |
|----------------------|----------------------|----------------------|----------------------|--------------------|
| \(\|err\|_2\) | \(\|err\|_\infty\) | \(\|err\|_1\) | \(\|err\|_1\) | \(\|err\|_1\) | \(\|err\|_1\) | \(\|err\|_1\) | \(\|err\|_1\) | MC (10^3 times) |
| DNN | 0.0192 | 0.0155 | 0.0202 | 0.0121 | 0.0248 | 0.0133 | 0.0416 | 0.0244 | 64746 |

TABLE 9. The feasible solutions of optimization.

| Optimization framework | \(R_{\text{ToC}}(\%)\) | \(R_{\text{mc}}(\%)\) | \(R_{\text{mc}}(\%)\) | \(R_{\text{mc}}(\%)\) | \(R_{\text{mc}}(\%)\) | \(R_{\text{W}}(\%)\) | \(R_{\text{W}}(\%)\) |
|-----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| Origin                | 51.00                | 51.14                | 49.19                | 49.78                | 1                    |
| Deterministic         | 1                    | 100                  | 99.90                | 0.95                 | 0.95                 |
| Fuzzy                 | 1                    | 100                  | 99.99                | 0.95                 | 0.95                 |
| Fuzzy                 | 2                    | 100                  | 99.94                | 0.95                 | 0.95                 |
| Fuzzy                 | 3                    | 100                  | 99.96                | 0.95                 | 0.95                 |
| Fuzzy                 | 4                    | 100                  | 99.26                | 100                  | 0.916               |
| Fuzzy                 | 5                    | 100                  | 99.41                | 100                  | 0.95                |
| Fuzzy                 | 6                    | 100                  | 99.94                | 100                  | 0.907               |
| Fuzzy                 | 7                    | 100                  | 99.24                | 100                  | 0.97               |
| Fuzzy                 | 8                    | 100                  | 97.89                | 100                  | 0.972               |

C. SOLUTIONS OF THE OPTIMIZATION MODEL

Applying the DNN-based surrogate models, the optimization process is carried out by Elitist Non-dominated Sorting Genetic Algorithm (NSGA2) [44]. The termination condition of NSGA2 is set to reach the maximum iteration number. The final iteration respectively captures final Pareto fronts of two optimization models, as shown in Fig. 13. The following images indicate that it is an approximately inverse proportional relationship to describe the trade-off between \(\lambda\) and \(\beta\). According to the computing time in Table 7, performing MC simulation would cost 3237 sec for one individual to calculate the reliability-based fitness. Thus, it is easy to estimate the total time cost would be approximately 187.3 days to complete the optimization under the given parameter setting (population size = 25, maximum iteration number = 200). However, these two optimization procedures respectively cost 4.38 min and 4.45 min using the DNN-based surrogate models.

Uniformly along the two Pareto fronts, eight Non-dominated cycle solutions are respectively picked for assessment. These cycle solutions are respectively input to the thermodynamic-based aeroengine model to perform the MC simulations. Then, the simulation data of overall performances is utilized to calculate the reliability of each operating state. Table 9 shows the comprehensive assessment of all cycle solutions, including weight ratio and reliability results. The results demonstrate that the original cycle scheme only achieves approximately 50% reliability for all operating states. While all optimized solutions make the thrust reliability surpassing 98% and SFC reliability reaching to a minimum of 92.87%, which shows satisfactory reliability. Observed the \(R_{\text{W}}\) of all solutions, it is found that the deterministic optimization strictly adheres to the constraint and rejects the relevant unfeasible solutions. While the fuzzy optimization mildly violates the constraint and generates a solution whose \(R_{\text{W}} = 0.916 > 0.9\).

Another investigation is implemented to analyze the detailed information of all optimized Pareto solutions, and the corresponding results are shown in Table 10. Compared to the original cycle, many cycle parameters of optimized cycles are improved to create more performance redundancy, which are in line with our expectations. However, it is disappointing to find that the cycle solutions of deterministic optimization are extremely similar. This similarity is reflected in the very narrow numerical ranges of each cycle parameter. For instance, the ranges of \(T_{\text{ET}}\) and \(R_{\text{BP}}\) are respectively \([1662.4, 1665.2]\) and \([5.321, 5.332]\). These two parameters are fixed so that the quantity of available choices has been substantially diminished. Thus, it means that the deterministic optimization makes a limited contribution to the flexibility in the design. However, a wider flexible design universe is provided by all the cycle alternatives from the fuzzy optimization. Obviously, the numerical ranges of these cycle parameters are much larger. Comparatively, the ranges of \(T_{\text{ET}}\) and \(R_{\text{BP}}\) are respectively \([1639.8, 1654.2]\) and \([5.239, 5.409]\). The same situation also goes for other cycle parameters. The above situation indicates that the fuzzy optimization is more capable of generating multiple feasible cycle alternatives with diversity.
A case study is carried out to further compare the two minimum $T_{ET}$ proposals presented in Table 9. According to the results of MC simulation in Table 8, both two solutions are verified to achieve the expected reliability for all operating conditions. Besides, the most striking difference between these two solutions is in $T_{ET}$. In particular, the $T_{ET}$ of deterministic optimum solutions is 1662.4K while the $T_{ET}$ of fuzzy optimum solution is 1639.8K. The difference of $T_{ET}$ is 22.6 K, which is a large gap referring to the presentation in Introduction.

For each operating condition, the performance original data is presented in the form of empirical cumulative distribution function (CDF), as shown in Fig. 14. The observation is that all the empirical CDF curves of optimized solutions are in the place that reach higher reliability value, referring to the benchmark lines. Nevertheless, all the thrust CDF curves of deterministic optimum solution are moved further than that of fuzzy optimum solution. This means that deterministic optimum solution causes excessive thrust redundancy to reach the expected reliability at the price of higher $T_{ET}$, which also reduces SFC redundancy.

For fuzzy optimization, the reason for the preference for moderate solutions is related to its objective functions. These objective functions are formulated by improved Z-shaped membership function specified by different $q(j)$. As presented in Eq. (31), each $q(j)$ is set to be less than 1. This value setting is in accordance with the law of diminishing marginal utility [45]. To be specified, this law makes the increment of satisfaction decay as each unit increase of objective function fitness. It effectively suppressed NSGA2’s strong willingness of minimizing all fitness value by radically increasing cycle parameter values. However, this willingness is fulfilled in deterministic optimization and causes the above excessive increases and the tendency that all solutions converge to be the same. While fuzzy optimization maintains the diversity in solutions and makes the cycle parameters are rationally increased to avoid unnecessary performance waste.
VI. CONCLUSION

This study proposes the feasible performance reliability design methodology using the DNN-based surrogate models. For enhancing the DNN performance, MINLP-based hyperparameter tuning technique is presented to obtain the optimum hyperparameter combination. With the application of well-tuned DNN model, the presented fuzzy optimization is implemented to design the commercial aeroengine thermal cycle under the consideration of uncertainty. The main conclusions of this study are presented as follows:

1. The proposed MINLP-based hyperparameter tuning technique can acquire the optimal hyperparameter combination of the specified DNN automatically and efficiently. This technique frees the programmer from the heavy work of hyperparameter manual adjustment, and lays the solid foundation for the DNN model to effectively execute a particular deep learning task.

2. Based on well-tuned hyperparameters, the trained DNN-based surrogate models could achieve high accuracy for comprehensive reliability prediction with negligible computing time cost (about 0.003% as that of MC simulation). These two characteristics enable DNN-based surrogate models to replace the time-consuming probabilistic analysis based on MC simulation so that facilitate the reliability-based optimization.

3. The optimization results indicated that deterministic optimization is easy to result in homogeneity of solutions. While the presented fuzzy optimization can acquire a variety of feasible solutions, which could reach the satisfied reliability (>92.87%) with diversity. Providing a variety of feasible candidate proposals could also assist the decision-maker to select wisely according to practical engineering situations.

4. The application of improved Z-shaped membership function could efficiently regulate the interrelations of each feasible objective. That regulation contributes to producing moderate solutions, which are rationally increasing the cycle parameters for creating reasonable performance redundancy. Thus, it reduces the potential overall performance waste, and conduces to deal with risks related to technical issues and aeroengine comprehensive cost.
CONFLICT OF INTEREST STATEMENT
The authors declare that there is no conflict of interests regarding the publication of this article.

NOMENCLATURE
Alt altitude (ft)
Amb ambient atmospheric condition
Cool cooling stream
Dtrain training set
err vector of errors
Ft net thrust (kN)
f fuel
HP high pressure spool
ISA international standard atmosphere
J Jacobi matrix
LR learning rate
LP low pressure spool
M Mesh
N relative rotating speed of spool (%)
NHL number of hidden layers
NNe number of neurons
P power (kW)
Pamb ambient atmospheric pressure (kPa)
Pt total pressure (kPa)
q shape factor
R reliability
RBP bypass ratio
RW weight ratio
r random number
S scaling factor
SFC specific fuel consumption (kg/(h*kN))
Tamb ambient atmospheric temperature (K)
TET turbine entry temperature (K)
Tt total temperature (K)
W mass flow
w objective weight

A. GREEKS
σ standard deviation
η efficiency
λ the first objective value
β the second objective value
µ satisfaction degree
π pressurization rate
ψ decision element
θ network hyperparameter
Δ mesh size
τ objective value

B. SUBSCRIPT
cor corrected parameter
u upper numerical boundary
l lower numerical boundary
std standard state

REFERENCES
[1] R. Kurz and K. Brun, “Degradation in gas turbine systems,” J. Eng. Gas Turbines Power, vol. 123, no. 1, pp. 70–77, Jan. 2001, doi: 10.1115/1.1340629.
[2] J. Qin, W. Zhou, W. Bao, and D. Yu, “Thermodynamic analysis and parametric study of a closed Brayton cycle thermal management system for scramjet,” Int. J. Hydrogen Energy, vol. 35, no. 1, pp. 356–364, Jan. 2010, doi: 10.1016/j.ijhydene.2009.09.025.
[3] K. Cheng, J. Qin, H. Sun, C. Deng, C. Lv, S. Zhang, and W. Bao, “Thermodynamic analysis for high-power electricity generation systems based on closed-Brayton-cycle with finite cold source on hypersonic vehicles,” Int. J. Hydrog. Energy, vol. 1, pp. 1–13, May 2018, doi: 10.1016/j.ijhydene.2018.05.138.
[4] N. Cumpsty and A. Heyes, “The principle and layout of jet engines,” in Jet Propulsion. Cambridge, U.K.: Cambridge Univ. Press, 2003, ch. 5, sec. 1, pp. 56–58.
[5] W. Yao, X. Chen, W. Luo, M. van Tooren, and J. Guo, “Review of uncertainty-based multidisciplinary design optimization methods for aerospace vehicles,” Prog. Aerosp. Sci., vol. 47, no. 6, pp. 450–479, Aug. 2011, doi: 10.1016/j.paerosci.2011.05.001.
[6] P. Yu, D. Zhang, Y. Ma, and J. Hong, “Dynamic modeling and vibration characteristics analysis of the aero-engine dual-rotor system with fan blade out,” Mech. Syst. Signal Process., vol. 106, pp. 158–175, Jun. 2018.
[7] D. M. Frangopol and K. Maute, “Life-cycle reliability-based optimization of civil and aerospace structures,” Comput. Struct., vol. 81, no. 7, pp. 397–410, Apr. 2003, doi: 10.1016/S0045-7949(03)00020-8.
[8] M. Chen, J. Zhang, and H. Tang, “Interval analysis of the standard of adaptive cycle engine component performance deviation,” Aerosp. Sci. Technol., vol. 81, pp. 179–191, Oct. 2018, doi: 10.1016/j.ast.2018.07.004.
[9] Y. Wei, G. Bai, and L.-K. Song, “A novel reliability analysis approach with collaborative active learning strategy-based augmented RBF metamodel,” IEEE Access, vol. 8, pp. 199603–199617, 2020, doi: 10.1109/ACCESS.2020.3055670.
[10] K. L. Nylund, T. Asparouhov, and B. O. Muthén, “Deciding on the number of classes in latent class analysis and growth mixture modeling: A Monte Carlo simulation study,” Struct. Equation Model. A, Multidisciplinary J., vol. 14, no. 4, pp. 535–569, Dec. 2007, doi: 10.1080/10705510701575396.
[11] G. Bai and C. Fei, “Distributed collaborative response surface method for mechanical dynamic assembly reliability design,” Chin. J. Mech. Eng., vol. 26, no. 6, pp. 1160–1168, Nov. 2013, doi: 10.3901/CJME.2013.06.1160.
[12] B.-W. Wang, W.-Z. Tang, L.-K. Song, and G.-C. Bai, “PSO-LSSVR: A surrogate modeling approach for probabilistic flutter evaluation of compressor blade,” Structures, vol. 28, pp. 1634–1645, Dec. 2020, doi: 10.1016/j.istruc.2020.10.007.
[13] D. Cao and G. Bai, “A study on aeroengine conceptual design considering multi-mission performance reliability,” Appl. Sci., vol. 10, no. 13, p. 4668, Jul. 2020, doi: 10.3390/app10134668.
[14] L.-K. Song and G.-C. Bai, “Multi-surrogate collaboration approach for creep-fatigue reliability assessment of turbine rotor,” IEEE Access, vol. 8, pp. 39861–39874, 2020, doi: 10.1109/ACCESS.2020.2975316.
[15] L.-K. Song, C.-W. Fei, J. Wen, and G.-C. Bai, “Multi-objective reliability-based design optimization approach of complex structure with multi-failure modes,” Aerosp. Sci. Technol., vol. 64, pp. 52–62, May 2017, doi: 10.1016/j.ast.2017.01.018.
[16] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, Jan. 2015, doi: 10.1038/14539.
[17] G. Martin, V. Talon, P. Higelin, A. Charlet, and C. Caillol, “Implementing turbomachinery physics into data map-based turbocharger models,” SAE Int. J. Engines, vol. 2, no. 1, pp. 211–229, Apr. 2009, doi: 10.4271/2009-01-0310.
[18] D. Mavris, N. Maccotail, and B. Roth, “A probabilistic design methodology for commercial aircraft engine cycle selection,” in Proc. Aviation Congr. Expo., Anaheim, CA, USA, Sep. 1998, pp. 1432–1441, doi: 10.4271/985510.
[19] D. Mavris, B. Roth, and N. Maccotail, “A method for probabilistic sensitivity analysis of commercial aircraft engines,” in Proc. 14th Int. Soc. Air Breathing Engines Conf., Florence, Italy, Aug. 1999, doi: 10.1002/9780470999455.ch1.
[20] M. Chen, K. Zhang, and H.-L. Tang, “A probabilistic design methodology for a turboshaft engine overall performance analysis,” Adv. Mech. Eng., vol. 6, Jan. 2014, Art. no. 976853, doi: 10.1155/2014/976853.
References

[21] F. Peyraut, Z.-Q. Feng, and N. Labed, “A general convergence condition of the Newton-Raphson algorithm applied to compressible hyperelasticity,” Struct. Eng. Mech., vol. 21, no. 2, pp. 121–136, Sep. 2005, doi: 10.12989/seme.2005.21.2.121.

[22] J. Zheng, M. Chen, and H. Tang, “Matching mechanism analysis on an adaptive cycle engine,” Chin. J. Aeronaut., vol. 30, no. 2, pp. 706–718, Apr. 2017, doi: 10.1016/j.ijjamech.2017.02.006.

[23] J. Zheng, H. Tang, M. Chen, and F.-J. Yin, “Equilibrium running principle analysis on an adaptive cycle engine,” Appl. Thermal Eng., vol. 132, pp. 393–409, Mar. 2018, doi: 10.1016/j.applthermaleng.2017.12.102.

[24] J. Zhang, H. Tang, and M. Chen, “Linear substitute model-based uncertainty analysis of complicated non-linear energy system performance (case study of an adaptive cycle engine),” Appl. Energy, vol. 249, pp. 87–108, Sep. 2019, doi: 10.1016/j.apenergy.2019.04.138.

[25] T. Serre, G. Kreiman, M. Koulh, C. Cadieu, U. Knoblich, and T. Poggio, “A quantitative theory of immediate visual recognition,” Proc. Brain Res., vol. 165, pp. 33–56, Oct. 2007, doi: 10.1016/S0068-1229(06)60048-4.

[26] Y. Yoo, “Hyperparameter optimization of deep neural network using univariate dynamic encoding algorithm for searches,” Knowl.-Based Syst., vol. 178, pp. 74–83, Aug. 2019, doi: 10.1016/j.knosys.2019.04.019.

[27] V. Strijov and G. W. Weber, “Nonlinear regression model generation using hyperparameter optimization,” Comput. Math. Appl., vol. 60, no. 4, pp. 981–988, Aug. 2010, doi: 10.1016/j.camwa.2010.03.021.

[28] M. Wustuba, N. Schilling, and L. Schmidt-Thieme, “Learning hyperparameter optimization initializations,” Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSSA), Paris, France, Oct. 2015, pp. 1–10, doi: 10.1109/DSSA.2015.7344817.

[29] S. Jeyanthi and M. Subadra, “Implementation of single neuron using various activation functions with FPGA,” in Proc. IEEE Int. Conf. Adv. Commun., Control Comput. Technol., Ramanathapuram, India, May 2014, pp. 1126–1131, doi: 10.1109/ICACCT.2014.7019273.

[30] C. Audet and J. E. Dennis, “Mesh adaptive direct search algorithms for constrained optimization,” SIAM J. Optim., vol. 17, no. 1, pp. 188–217, Jan. 2006, doi: 10.1137/040603371.

[31] S. Le Digabel, “Algorithm 909: NOMAD: Nonlinear optimization with multiobjective genetic algorithm: NSGA-II,” IEEE Trans. Evol. Comput., vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996071.

[32] D. R. Just and H. H. Pettersson, “Diminishing marginal utility of wealth and calorie and calibration of risk in agriculture,” Am. J. Agricult. Econ., vol. 85, no. 5, pp. 1234–1241, Dec. 2003, doi: 10.1111/j.0002-5853.2003.00536.x.

---

DALU CAO was born in Hunan, China, in 1990. He received the B.S. degree in flight vehicle propulsion engineering from Harbin Engineering University, in 2014, and the M.S. degree in aviation engineering from Beihang University, in 2017, where he is currently pursuing the Ph.D. degree with the school. His research interests include aeroengine performance design and analysis, optimization algorithm, and machine learning.

GUANG-CHEN BAI was born in Harbin, China, in 1962. He received the B.S. degree in mechanical design and theory from the Harbin University of Science and Technology, China, in 1983, and the Ph.D. degree in structural reliability assessment from the Harbin Institute of Technology, China, in 1993. From 1993 to 1995, he was an Assistant Professor with the School of Aeronautics, Harbin Institute of Technology. From 1995 to 2000, he was an Associate Professor with the School of Energy and Power Engineering, Beihang University, where he was a Professor with the School of Energy and Power Engineering, from 2000 to 2016. Since 2016, he has been a tenured Professor with the School of Energy and Power Engineering, Beihang University. He is the author of more than 100 articles. His research interests include fatigue reliability assessment, vibration reliability evaluation, and multidisciplinary design optimization. He was a member of the American Institute of Aeronautics and Astronautics (AIAA) and the Chinese Society of Aeronautics and Astronautics (CSAA). He is a Reviewer of more than 20 journals, such as Chinese Journal of Aeronautics, Reliability Engineering and System Safety, and Chinese Journal of Mechanical Engineering.