Proceedings of ASME Turbo Expo 2023
Turbomachinery Technical Conference and Exposition
GT2023
June 26-30, 2023, Boston, Massachusetts

GT2023-102054

AXIAL COMPRESSOR MAP GENERATION LEVERAGING AUTONOMOUS SELF-TRAINING ARTIFICIAL INTELLIGENCE. PHASE 2

Maksym Burlaka
SoftInWay, Inc.
Burlington, MA

Sascha Podlech
SoftInWay, Inc.
Burlington, MA

Leonid Moroz
SoftInWay, Inc.
Burlington, MA

ABSTRACT

This paper discusses a study performed by SoftInWay as part of a Phase II SBIR project funded by NASA.

In contrast with the Phase I project (published in paper GTP-22-1328 [1]) where three discrete compressors were considered, the Phase II study was focused on addressing the problem of axial compressor long development time and cost with the use of AI models capable of predicting the geometry and performance of various multi-stage axial compressors with multiple variable vanes. The applicability of the AI models to various compressors enables the opportunity to avoid iterations between engine cycle analysis and compressor design.

In this paper, automated compressor design and performance generation workflows are described. The approach for autonomous selection of the architectures and hyperparameters of Machine Learning (ML) models is explained. The uncertainty quantification techniques are considered. The developed ML-powered methods for compressor geometry prediction are discussed. The ML models' accuracy values and representations of typical geometry and performance predictions are given. The utilization of the ML models in engine cycle analysis is discussed.

Keywords: Axial Compressor, Performance Map, Machine Learning, Artificial Neural Network, Compressor Vanes' Angles, Optimization

NOMENCLATURE

NASA	The National Aeronautics and Space Administration
SBIR	Small Business Innovation Research
AI	Artificial Intelligence
ML	Machine Learning
SHM	Sequential Hybrid Model Approach
1D	One-Dimensional
2D	Two-Dimensional
3D	Three-Dimensional
CFD	Computational Fluid Dynamics
NN	Neural Network
AutoML	Automated Machine Learning
NPSS	Numerical Propulsion System Simulation
EEE	Energy Efficient Engine
LPC	Low-Pressure Compressor
HPC	High-Pressure Compressor
LPT	Low-Pressure Turbine
HPT	High-Pressure Turbine
DCA	Double Circular Arc
IGV	Inlet Guide Vane
VGV	Variable Guide Vane
RPM	Revolutions Per Minute
RD	Relative Difference [%]
MSE	Mean Squared Error
MFR	Mass Flow Rate [kg/s]
ptr	Total-to-Total Pressure Ratio
psr	Total-to-Static Pressure Ratio
eff_tt	Total-to-Total Efficiency
eff_ts	Total-to-Static Efficiency
mfr_dp	Mass Flow Rate at the Design Point [kg/s]
ptr_dp	Total-to-total Pressure Ratio at the Design Point
Ut	Rotor Tip Speed [m/s]
CzU	Flow Factor
dbt	Hub-to-Tip Ratio of Diameters at the Inlet
LuUU_ave	Stage-Averaged Work Coefficient
LuUU_max	Maximal Work Coefficient
dsp	Specific Diameter Type. It is a categorical parameter that can be either equal to 0, 1, or 2: 0 corresponds to constant hub diameter, 1 corresponds to constant mean diameter, 2 corresponds to constant tip diameter
Cmz_Cm1	Meridional Velicities Gradient
a_n	Coefficient between 0 and 1, the index n = 2, 3, 4, 5, and 6 stands for VGV row number

Cd	List of input parameters for design models
Cod	List of input parameters for performance models
G	Geometrical parameters of the multistage axial compressor flow path
g	Encoded geometrical parameters of the multistage axial compressor flow path

1. INTRODUCTION

Modern gas turbines are always chasing a higher cycle efficiency and lower fuel consumption. To reach these goals, low and high-pressure compressors (HPCs) are constantly being pushed to achieve higher pressure ratios. Increasing the compressor pressure ratio results in a higher rotor tip relative Mach number in the HPC front stages and consequently, steeper performance characteristic maps. The compressors with steep characteristics typically require variable geometry inlet guide vanes as well as variable stators in the first few stages to provide the desired performance and stability in an engine. The design and development time of a modern high-pressure compressor with variable geometry can take years with many design-build-test iterations which involve testing many possible reset angles of the variable vanes. Determining the optimal combination of vane angle resets that will provide the desired compressor performance in an engine's system is a time-consuming and expensive step in developing high-pressure compressors.

The task of predicting with sufficient accuracy of the specific compressor performance at various combinations of IGV and multiple VGV was completed in the Phase I project [1]. The relative error of compressor performance prediction with the developed, in Phase I, AI-based approach was below 3 % for validation and test data sets. The approach developed in Phase I and its relation to the previous work in the area of prediction of compressor performance maps with ML [2-4] was discussed in [1]. Despite the advantages of the developed AI-based approach it was still required to have a detailed compressor design and perform respective performance data generation, and training for every new compressor. It would be more valuable to avoid training a new model for every new compressor, but instead, use a single model capable of predicting the performance of various multistage axial compressors accounting for variable IGV and multiple VGVs.

In the case of performance prediction of the specific compressor with VGVs, the list of input parameters is clear. Indeed, it is simply required to use the combinations of restagger angles for IGV and all VGVs, rotational speed, and mass flow rate as input parameters, because the compressor geometry is implicitly given. However, this is not the case if one wants to have a single model for the prediction of the performance of an arbitrary multistage axial compressor. A limited but sufficient number of compressor parameters shall be determined to make practical the task of developing of AI model capable to predict the map of the arbitrary compressor. Azzam [5] applied artificial neural networks for turbomachinery design. The authors were not aware of Azzam's work during the execution of work performed in Phase I and Phase II. Essentially Azzam with co-authors were following the same target. They considered 1100 different single-stage axial compressors, which were based on 5 compressor rigs. However, variable IGVs and VGVs were not taken into account and only basic geometrical parameters for single-stage axial compressors were considered, which diminished the practical utility of their model. In the current research, the authors considered a much wider and more complex task of multistage axial compressors with variable IGV and multiple VGVs leveraging the technology from Phase I. The methods utilized in this research are different compared to the ones applied by Azzam. It should be noted that at the moment of writing this paper, the Phase II project was still in progress. Thus, this paper presents the results of the R&D to date.

Figure 1 shows a schematic structure of the proposed idea. A compressor geometry generator produces various compressor designs and performance maps. The compressor's geometrical dimensions are reduced to a minimum number of parameters needed for the AI model. The datasets are used for training purposes of the AI model which in the end is utilized in a system-level analysis tool. Additionally, utilizing the model, the compressor geometry can be reconstructed if desired.

The proposed innovation is flexible regarding what level of fidelity can be used for the dataset generation. 1D solvers are fast and can deliver a vast amount of data in a short time. However, the peculiarity of 1D codes is that they are typically based on some empirical correlations or analytical equations derived from certain assumptions. On the other hand, 3D CFD methods could be used instead, which increases the fidelity but requires more resources and time. The end user can decide on which level of fidelity is required for the given task.

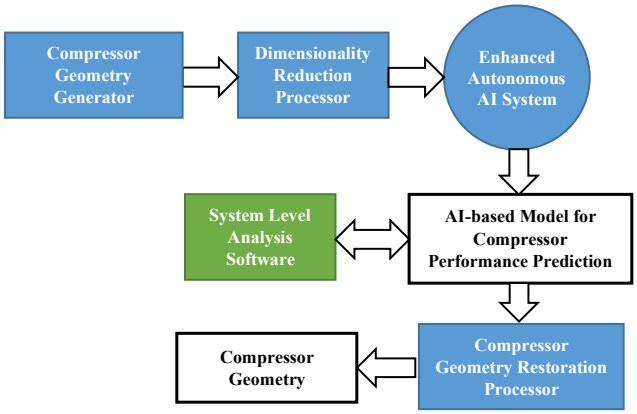


FIGURE 1: HIGH-LEVEL STRUCTURE OF THE PROPOSED TECHNOLOGY

The scope of the NASA Phase II project also included: the rig tests of the three-stage axial compressor at various reset angles of IGV by Maurice J. Zucrow Laboratories (Purdue University) for experimental evaluation of the AI models; and assessment of the performance of engine cycles using NPSS and AI models of axial compressors by Raytheon Technologies Research Center. However, at the moment of writing this paper, their work has not been completed yet and thus is not presented.

It should be noted that the proposed technology is not limited to axial compressors in general and can be tailored to other turbomachinery and non-turbomachinery components. However, taking into account that current research was focused on multistage axial compressors, the current implementation cannot be directly used for the other components. Besides, in this research, only air compressors were considered. The technology allows the creation of AI models for axial compressors operating with other working fluids, but it would require generating new datasets and training new models. Generalization of the models to various fluids without the need to train new models is the subject of future research.

2. MATERIALS AND METHODS

The following technical activities were performed:

- Input parameters determination
- Training data generation and pre-processing
- ML techniques, hyperparameters fine-tuning, and training
- AI model testing in the cycle analysis tool

This section describes the approaches and methods used for every type of technical activity. The tools used for this study are part of the AxSTREAM® platform and include workflow automation software and generative Design methods which create a compressor flow path geometry based on a certain amount of input parameters [6].

2.1 Input Parameters Determination

Concerning input parameters determination for the universal AI model for multistage axial compressors, the first intuitive choice is to explicitly use the entire compressor flow path geometry as input data to predict its performance. However, the definition of the exhaustive list of flow path geometry for a multi-stage axial compressor is not a straightforward task. For example, one performance code (1D/2D) requires one set of parameters to define the flow path of the entire compressor, another performance code requires a different set of parameters depending on the calculation methods, models, program implementations, and even preferences of the developers. Of course, there are similar parameters, like diameters, stage number, blade heights, angles, etc. however the complete set is never going to be the same. Which of these parameters should be used as a vector of input data to define the geometry? Another option is to use some universal file format that can contain the geometry of the entire compressor, such as STEP [7] or CGNS [8]. However, in this case, it is required to deal with millions of plain geometrical coordinates of the surfaces defining the compressor flow path geometry. Moreover, if one decides to use the AI model of the compressor in cycle simulation it would be required to feed the entire compressor geometry to the model to predict its performance and at the cycle analysis step it is often not available and if available then performance maps are already accessible and can be directly used in the cycle or via AI model trained employing Phase I approach. Taking all this into account it becomes clear that, it is not reasonable to use the entire geometry of the compressor as an input to predict its performance by an AI model. Moreover, the AI models should be independent of any specific performance codes and be compatible with various sources of performance and geometry data.

For convenience, the set of input parameters to be used by the end-user of the model was designated as **C**. The parameters included in **C** were selected to be sufficient to unequivocally define the compressor geometry and its performance considering the availability of the parameter to a user either at the system level or the value of the parameter can be set to some default value unless a user decides to change it by him/herself. It should be noted that for design and off-design the lists of parameters are different. The list of input parameters for design was designated as **Cd** and the list of input parameters for performance (off-design) was designated as **Cod**. The parameters included in **Cd** and **Cod** along with the respective ranges are shown in Table 1 ('+' means that the parameter is included in the list, and '-' means that the parameter is not included in the list). As can be seen **Cod** includes **Cd** but also includes IGV/VGV reset angles, shaft rotational speed, and mass flow rate. Most of the selected parameters are dimensionless, except for mfr_dp and Ut. **Cd** parameters are not difficult to specify even at the conceptual stages of airbreathing engine development. The parameters additional to **Cd** are required to define the off-design point at the performance of the compressor to be determined at. **Cd** parameters will be used as an input vector to both the geometrical and performance models of the compressor. It should be noted that the inlet design point condition corresponds to standard conditions: total inlet temperature equals 288.15 K and total inlet pressure is 101.325 bar with the air as a working fluid, which makes it easy to convert the compressor performance to the desired inlet temperature and pressure using equations for corrected parameters [9]. The ranges for parameters (Table 1) were selected to cover typical values used for axial compressors in the industry. Moreover, at this step, the authors were mostly focused on the approach itself and not on the development of the most universal model applicable to every imaginable compressor design. The idea was to develop an approach and demonstrate its feasibility on various compressors within the considered ranges of parameters. Once developed, the approach can be applied to the extended range and/or tailored to the specific requirements.

The output parameters for the AI models include the flow path geometry of the compressor and the performance parameters such as ptr, eff_tt, psr, and eff_ts.

2.2 Training Data Generation and Pre-processing

In Phase I, the performance data generation was conducted with three manually designed compressors. Since the goal is to predict the performance of an arbitrary compressor with design parameters in the above-mentioned ranges, a vast amount of data is necessary. Additionally, the geometrical parameters for the reconstruction of the compressor include 497 values. These include the tip/hub diameter of each stage, chord values, etc., and are declared as **G** parameters. The multistage axial compressor flow path geometry creation and extraction of these values for a

vast amount of compressor designs are not feasible manually, therefore an automated workflow was created within an automation software. A schematic workflow is depicted in Figure 2a. The Sobol sequence [10] is used to generate various combinations of the input parameters. Each combination undergoes the following procedure. First, the selected input parameters are used in the data preparation step to calculate additional variables necessary in the 1D Generative Design tool like rotational speed, number of stages, etc. Based on this data, the design calculation within the 1D Generative Design tool is then performed. In the following steps, the stagewise work distribution is determined based on the input value for the average work coefficient.

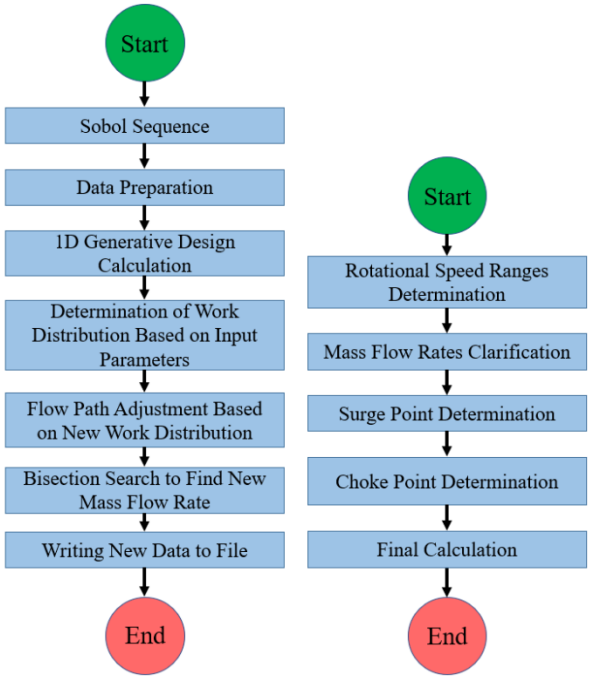
TABLE 1: AI MODEL INPUT PARAMETERS

Parameter	Range		Cd	Cod
	min	max		
mfr_dp	6.0	150.0	+	+
ptr_dp	1.2	22.0	+	+
Ut	140	540	+	+
CzU	0.25	0.7	+	+
dbt	0.4	0.9	+	+
LuUU_ave	0.13	0.45	+	+
LuUU_max	0.14	0.58	+	+
dsp	0	2	+	+
Cmz_Cm1	0.6	1.0	+	+
IGV	0.0	30.0	-	+
VGV2	0.0	IGV*a_2	-	+
VGV3	0.0	VGV2*a_3	-	+
VGV4	0.0	VGV3*a_4	-	+
VGV5	0.0	VGV4*a_5	-	+
VGV6	0.0	VGV5*a_6	-	+
rpm	3450	26400	-	+
MFR	3	176	-	+

Once the distribution of specific work at the mean section is calculated it is written to the respective compressor project file where the flow path is automatically adjusted taking new specific work distribution into account. It should be noted that after the adjustment procedure, the pressure ratio might differ from the design pressure ratio. Therefore, the search for the mass flow rate value which will bring the pressure ratio to the design one is performed by the bisection search algorithm.

Once a new mass flow rate value is found, DCA profiles are applied to all blade rows in subsequent procedures to allow better calculation with transonic blades. After that, the obtained compressor flow path geometry **G** is written to a file that will be used as geometrical data set for ML. This procedure is repeated for the other combinations of input parameters provided by the Sobol sequence.

The off-design data generation and pre-processing are performed according to methods developed in Phase I and will be briefly summarized here. Detailed information regarding the data generation workflow and parametric models for the representation of speedlines and efficiency lines can be found in the paper by Burlaka [1]. A schematic workflow of the procedure is shown in Figure 2b. The main purpose of the automated workflow is to determine the stable operating range of the analyzed compressor which includes removing outliers, smoothing curves, etc., since it is crucial to provide good and clean data to the training algorithm. Every block in the shown figure contains a sub-workflow that will not be explained in detail. The shown process is performed for various IGV/VGV combinations. The first step is the determination of rotational speeds for which the compressor can operate. Afterward, the mass flow ranges and surge/choke point will be determined. The surge point is usually characterized as the maximum static pressure ratio on a given speedline and the choke point location is based on the criteria when the efficiency drops by 15% relative to the maximum value on the speedline. In Phase II the same workflow was used. It was modified only to account for multiple compressor designs according to the set of input parameters.



a) b)

FIGURE 2: A) AUTOMATIC WORKFLOW FOR AXIAL COMPRESSOR FLOW PATH GEOMETRY DATA GENERATION; B) AUTOMATIC WORKFLOW FOR COMPRESSOR PERFORMANCE DATA GENERATION AND PRE-PROCESSING

During the process of generation of the data sets for compressors, authors developed a new parametric model-based approach for the representation of the compressor flow path geometry in meridional cross-section to improve the quality of prediction, because preliminary predictions of compressor flow path designs sometimes showed unfeasible geometries. One of the most extreme examples of such unfeasible geometries is depicted in Figure 3, where the hub diameters are greater than tip diameters, and overall diameters predictions are inconsistent. Such an effect is not a result of bad training it is the consequence

of the fact that the differences between the hub and tip diameters in high-pressure compressors are small compared to the values of the diameters, so even relatively small discrepancies in predictions could result in hub diameters being higher than tip diameters and overall the predictions of the models have to be extremely accurate to have the smooth meridional shape of the flow path. Such accuracy is practically impossible to achieve.

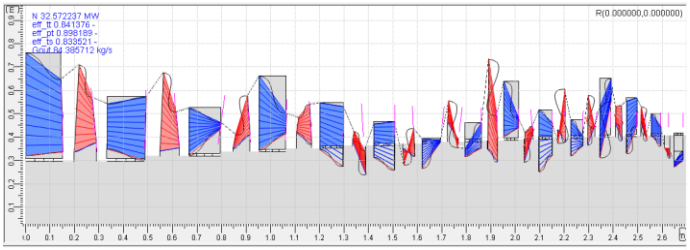


FIGURE 3: UNSUCCESSFULLY RECONSTRUCTED COMPRESSOR GEOMETRY FROM PREDICTION

In order to make the geometry predictions tolerant to inevitable imperfections of NN predictions, a Bezier curve [11] representation of the stagewise distribution of blade heights was implemented. It was decided to use the outlet diameter of the first compressor stage and the outlet diameter last stage as the anchor points and reconstruct the stagewise distribution of blade heights starting from them using the relative coordinates of the Bezier curve base points (Figure 4). The flexibility of the Bezier curve of the 2nd order was not sufficient and it was decided to use curves of the 3rd order which are defined by four base points P0, P1, P2, and P3. P1 and P2 are determined relative to anchor points P0 and P3, as follows:

$$P1_x = P0_x + \Delta a' \;;\; P2_x = P3_x - \Delta a$$
$$P1_y = P0_y + \Delta b' \;;\; P2_y = P3_y - \Delta b \quad (1)$$

where

$$\Delta a' = (P3_x - P0_x) \cdot a' \;;\; \Delta a = (P3_x - P0_x) \cdot a$$
$$\Delta b' = (P3_y - P0_y) \cdot b' \;;\; \Delta b = (P3_y - P0_y) \cdot b \quad (2)$$

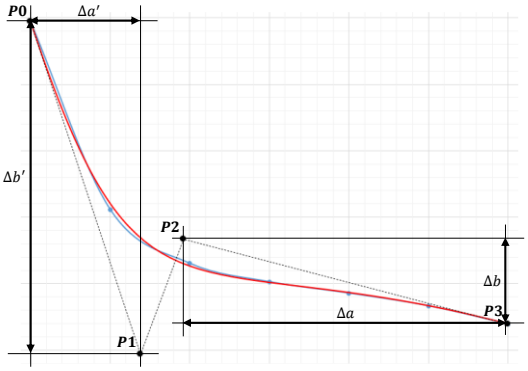


FIGURE 4: EXAMPLE OF THE PARAMETRIC CURVE FOR COMPRESSOR FLOW PATH GEOMETRY

The utilized 1D/2D software uses geometrical parameters like axial creance only on the hub. The effort of this study focuses on the meanline section. Therefore, a method needed to be found in which the leading and trailing edge position of the blades on the mean section can be determined to ensure the correct gap size between blades and ultimately the length of the entire machine. This led to the addition of coefficients K and K_o which are defined as follows:

$$K = \frac{B_h}{B_m} \quad (3)$$

$$K_O = \frac{mean\ offset\ inlet}{B_m} \quad (4)$$

B_h is the axial chord length on the hub section and B_m is the axial chord length at the mean section defined as:

$$B_m = c \cdot cos(\gamma) \quad (5)$$

where c is the blade chord and γ is the stagger angle.

The mean offset inlet value is the distance between the hub leading edge axial position and the mean section leading edge position. It is illustrated in Figure 5. These values, as well as the hub axial chord length, were extracted by a Python script after the designs were created.

These coefficients have as input the stage number and also Bezier-Curves of 3rd order with the same representation method as described above for the blade heights. The diameter values are excluded from training completely (except the first and last row) and the spline coefficients of each spline are now the input parameters for the models. Once a model is trained the process of determining the blade diameters is as follows:

1) K and K_O spline coefficients are predicted by the trained model
2) K and K_O coefficients are used to determine the axial outlet position of each blade by equations (4) and (5)
3) Height is extracted from the spline at the various axial blade outlet positions.

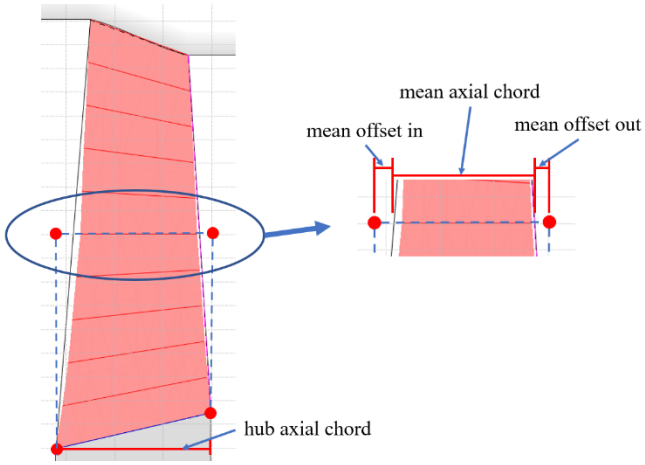


FIGURE 5: DEPICTION OF OFFSET PARAMETERS

It should be noted that the splines for 3-stage compressors do not have a sufficient number of points to fit accurately. A 3-stage compressor can only provide three points for the spline fitting. Since a 3rd order spline is used, four points are at least needed to build the spline. The conclusion was to have the

algorithm be trained for the K and K_O coefficients directly. For compressors with stages 4 and more, the spline approach was used.

By utilizing the spline and other definitions, the list of input parameters of 497 was reduced to 268 for 4-15 stages designs, and 83 for 2-3 stages.

Additionally, the dataset is divided into groups based on stage numbers. The reason for this is that in the previous approach the model tried to predict the entirety of parameters which led to situations where in the example of a 10-stage compressor, the model predicted the values for stage 11 up to stage 15 as well. This led to unreasonable values, e.g. stage 12 rotor blade count would equal 1. The reason for this is that the model tried to predict zeros for these values. Grouping the dataset removed this behavior.

The final training dataset includes 54,922 compressor geometries.

Several ways to represent predicted speedlines by the AI model exists. For example, polynomials of second or third order could be utilized, but they do not provide the required flexibility to represent the wide variety of different speedline shapes. Certain approaches even suffer fluctuations. The solution was found by combining three Bezier curves [12]: two linear and one quadratic curve for the middle section. The representation is shown in Figure 6. A modified parametric model was utilized for the efficiency plots. More details can be found in [1].

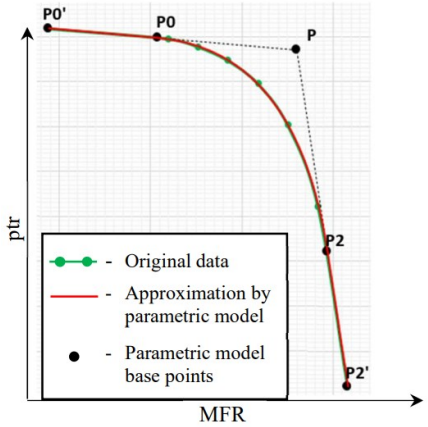


FIGURE 6: PARAMETRIC MODEL

2.3 ML Techniques, Hyperparameters Fine-tuning, and Training

Overall the approach for off-design models has not changed since Phase I [1]. The same parametric models for ptr, psr, eff_tt, and eff_ts and the structure of all NN models are used in Phase II. In Phase I it was shown that a single direct NN model for compressor performance prediction sometimes had oscillations not typical for compressor performance speedlines, and the parametric models' structure of 14 models was proposed: one NN model to predict choke MFR and one for surge MFR which were used for ptr, psr, eff_tt, and eff_ts curves plus 3 NN models to complete the prediction of ptr curve plus 3 NN models for psr curve plus 3 NN models for eff_tt curve, and 3 NN models for eff_ts curve. All these models were trained independently and then at the inference step the predictions of all the models were combined to recreate the entire speedlines for every considered performance parameter. It was found that every model has different architecture and set of hyperparameters to provide the best prediction accuracy. The data set was being extended gradually and from time to time it was required to perform training sessions of all 14 models on the continuously increasing data set. With that being said a manual search of NN architectures for all models was extremely labor-intensive and it was required to use automated techniques. It was determined that AutoML was able to find the NN architectures and hyperparameters resulting in a comparable or superior level of accuracy compared to the accuracy of manually engineered models. In Phase I in-house-developed AutoML was used, but in Phase II it was decided to switch to AutoKeras AutoML [13-15] due to a wider list of accounted hyperparameters and flexibility with tuners. It was possible to select a tuner among the following ones: Greedy, Bayesian, Hyperband, and Random. Preliminary investigations showed that the "greedy" tuner was typically providing more accurate models at the ceteris paribus. The maximal number of trials was set to 100 with a "greedy" tuner for automatic training of all performance models. The following list of hyperparameters was taken into account by AutoML during the search for the best NN model:

- Number of hidden layers
- Number of neurons
- Activation function
- Optimizer
- Learning rate
- Skip connections
- Dropout

An exhaustive description of the algorithms comprising AutoKeras is given in [13-15]. Author of [15] and one of the co-authors of [13, 14] – Haifeng Jin, summarizes the AutoKeras greedy search algorithm as follows [15]: "First, it iterates through a list of models to evaluate the target dataset. Second, it selects the current best model and builds a hyperparameter tree from it. The leaves in the tree are hyperparameters. Third, it generates a new hyperparameter value set by replacing the values of subtree leaves. The subtree is selected according to a probability distribution, which considers both the dependency relation between the hyperparameters and the size of the subtree. The less number of leaves a subtree contains, the more likely it is selected. Therefore, the search algorithm prefers the exploitation of the neighborhood of a good model to the exploration of new models. The new values are randomly generated. Fourth, go back to the second step and repeats until reach the maximum number of trials set by the user".

It should be noted that there are many other AutoML tools. Ferreira and others performed an extensive comparison of various AutoML tools in [16]. AutoKeras was not the best nor the worst AutoML tool for regression tasks. In turn, Ferreira concluded that AutoGluon [17] and H2O [18] are the best for

deep-learning regression tasks. Phase I approaches were based on the utilization of the TensorFlow ML platform with Keras and it was preferred to stay within the TensorFlow platform. In turn, AutoGluon and H2O required switching to alternative ML platforms and doing substantial changes to the techniques used in the project. Thus, the selection of AutoKeras for the Phase II project. The accuracy of the models provided by AutoKeras was further improved in the scope of this project as follows. The best model found by AutoKeras was trained 18 times starting from random initials for 10000 epochs each with early stopping to prevent overfitting. This enabled the opportunity to perform uncertainty quantification and also use the ensemble of 18 models for the prediction of each output parameter. Essentially every prediction of every output parameter is an average prediction of 18 respective models, which was usually more accurate than a single best model prediction. It was noticed that different layer weight initializers provide different qualities of uncertainty quantification. The analysis of the results of uncertainty quantification obtained with RandomUniform [19], GlorotUniform [20], and HeUniform [21] initializers showed that the HeUniform initializer provides the most consistent uncertainty quantifications. Uncertainty ranges obtained with HeUniform were widening when the model was extrapolating outside of the ranges or in the areas with a small density of points in the data set and narrowing when predicting the performance inside of the ranges with a good density of data points. Thus, the HeUniform initializer was used for the initializations of training of all models for uncertainty quantifications.

The combination of AutoKeras with the ensembles enabled the opportunity to have consistent quality of automatically optimized NN models assuring reproducibility and a solid foundation for future utilization of the developed methods by turbomachinery engineers without a need to be an ML expert. The direct comparison of the improved approach with the alternative AutoML tools was not performed, because it would require quite extensive resources and was not critical to the goals of the project. However, considering the results of the comparison of plain AutoKeras with H2O and AutoGluon from [16], the addition of ensembles should at least bring the accuracy closer to them if not achieve the same accuracy or higher. Note that H2O and AutoGluon use ensembles as well.

It should be noted that the quality of predictions of compressor performance speedlines and efficiency lines was determined as a relative difference (RD) calculated using equation (6) for every speedline and efficiency line from validation and test sets.

$$RD = \frac{2\sum_{i=1}^{n}(|A_{i-1} - P_{i-1}| + |A_i - P_i|) \cdot (G_i - G_{i-1})}{\sum_{i=1}^{n}(A_{i-1} + P_{i-1} + A_i + P_i) \cdot (G_i - G_{i-1})} \quad (6)$$

where,

A_i, A_{i-1} - the actual values of the considered parameter at the current G_i and previous G_{i-1} value of mass flow rate.

P_i, P_{i-1} - the predicted values of the considered parameter at the current G_i and previous G_{i-1} value of mass flow rate.

Then using the threshold of 5 % applied for RD the accuracy of the models was evaluated for combined validation and test sets using the equation (7). Typically accuracy is used in binary classification problems. However, applying the predefined error threshold for speedlines and efficiency lines enables the utilization of the accuracy concept for regression problems as well. When doing this, it is important to specify the RD threshold used for evaluation of model accuracy, e.g. $Accuracy_{5\%}$ or $Accuracy_{10\%}$, etc.

$$Accuracy_{5\%} = \frac{Predictions\ with\ RD < 5\ \%}{Total\ Number\ of\ Predictions} \quad (7)$$

For compressor flow path geometry predictions it was required to develop the techniques to achieve sufficient accuracy. Two approaches were considered. The first approach is direct NN trained to predict **G** by taking **Cd** as an input (Figure 7a). The second approach is the sequential hybrid model (SHM) shown in Figure 7b. SHM presumes execution of two steps. The first step is the training of the autoencoder [22]. The second step is the training of the model which takes **Cd** as an input and predicts **g** (autoencoder intermediate layer). Eventually, such a hybrid NN can predict **G** by taking **Cd** as an input as it was required. The training sessions for both cases were performed. It turns out that the MSE of the **Cd-g-G** model is consistently better compared to the MSE of the **Cd-G** model. Therefore, the SHM technique was used for the training of all geometrical models.

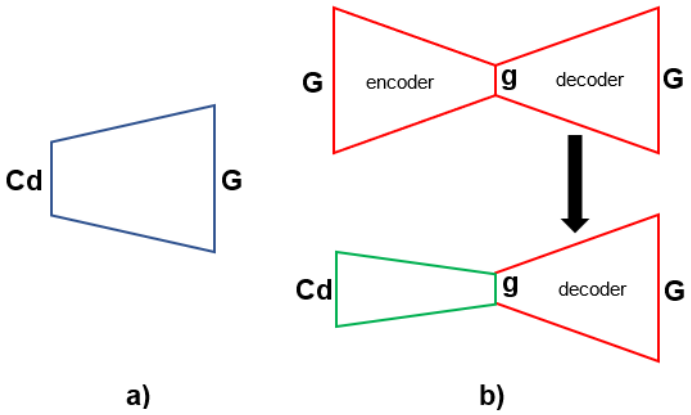


FIGURE 7: ARCHITECTURAL APPROACHES FOR COMPRESSOR GEOMETRY MODEL TRAINING: A) DIRECT MODEL; B) SEQUENTIAL HYBRID MODEL.

2.4 AI Model Utilization in The Cycle Analysis Task

Taking into account that full scope assessment of the performance of engine cycles using NPSS and AI models of axial compressors by Raytheon Technologies Research Center was not yet completed at the moment of writing this paper. For the sake of this paper, it was decided to demonstrate a simplified case of utilization of axial compressor AI models in the analysis of turbofan engines utilizing an in-house developed model of the engine within an automation software. The model of the engine and utilized methods and approaches were developed independently from the NASA Phase II project and were

presented at Turbine Engine Technology Symposium (TETS) in 2022 [23]. In this paper, all the details of the cycle simulation from [23] will not be described. Here only a high-level description of the cycle and the way the utilization of AI models are given.

The turbofan engine cycle is shown in Figure 8. It includes an intake, fan, low-pressure compressor (LPC), high-pressure compressor (HPC), combustor, high-pressure turbine (HPT), low-pressure turbine (LPC), and nozzle. In the original presentation from TETS, all the turbomachinery components were simulated using a commercially available Meanline/ Streamline solver without any AI models. However, in this paper, LPC and HPC are replaced with the AI models obtained in the scope of the NASA Phase II project. It should be noted that the same AI models are used for both compressors. The models were not specifically trained to predict the performance of LPC and HPC used in the original TETS presentation. The values of input parameters for the AI model were determined from the respective compressor 1D/2D projects and presented in Table 2. It should be noted that corrected values of mfr_dp and Ut were used as input to the AI model because the actual values of inlet pressure and inlet temperature for LPC and HPC were different from standard conditions used for performance data generation and then in the training of AI models. The current rpm and current MFR input values were corrected as well and they were coming to AI model input from the cycle level at every iteration during the convergence process. The current ptr and eff_tt values for both LPC and HPC along with respective choke and surge points were coming from the AI model to the cycle at every iteration during the convergence process. As can be seen from Table 2 the restagger angles for IGV, VGV2, and VGV3 were variable for HPC. The values for them were coming from the cycle level.

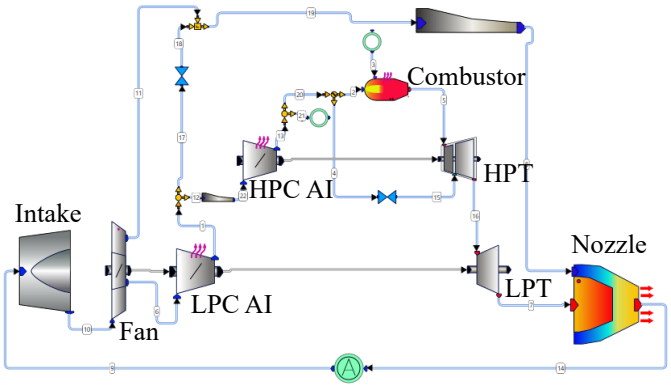


FIGURE 8: TURBOFAN ENGINE CYCLE REPRESENTATION WITH AI MODEL USED FOR LPC AND HPC PERFORMANCE DETERMINATION

In this paper, the goal of cycle analysis was to demonstrate the utilization of AI models of the compressor by running a single engine part-load point and evaluate the time of running the cycle with LPC AI and HPC AI models and compare it to the time of running a single engine part-load point using the original cycle from [23].

It should be noted that meridional velocities gradient Cmz_Cm1 for HPC (Table 2) is slightly higher than the maximal value of 1.0 in the data set (Table 1), i.e. the AI model was slightly extrapolating beyond the ranges for HPC.

3. RESULTS AND DISCUSSION

Leveraging the approaches and methods described in the previous section respective technical activities were performed. Selected results are presented and discussed in this section.

TABLE 2: AI MODEL INPUT PARAMETERS' VALUES FOR LPC AND HPC

Parameter	LPC	HPC
mfr_dp, kg/s	75.557	55.400
ptr_dp	1.320	18.660
Ut, m/s	249.0	412.0
CzU	0.571	0.480
dbt	0.666	0.457
LuUU_ave	0.224	0.304
LuUU_max	0.261	0.355
dsp	1	1
Cmz_Cm1	0.985	1.041
IGV, deg	0.0	var
VGV2, deg	0.0	var
VGV3, deg	0.0	var
VGV4, deg	0.0	0.0
VGV5, deg	0.0	0.0
VGV6, deg	0.0	0.0
Current rpm	var	var
Current MFR, kg/s	var	var

3.1 Compressor Geometry Prediction Results

After the final training session, the predictions of the geometry are evaluated and presented here. Figure 9 shows an overlay view of an original and a reconstructed compressor (from the test data set) geometry in the meridional section. One can notice that the two compressors have different lengths. This is due to the fact, that the reconstructed geometry only considers the mean line section, and no twist of the blades is included. This means that prismatic blades are created. The utilized 1D/2D program applies the axial blade clearance at the hub (as described above), hence the usage of prismatic blades results in different overall machine lengths. Therefore, the important aspects to look for here are the channel flow path, blade diameters, and mean section chord lengths which match the original geometry very well.

Figure 10 shows the blade profiles at the mean section of another compressor from the test set.

The comparison of the original inlet and outlet metal angles and the predicted ones is given in Table 3. The values of angles and the differences between them are given in degrees. As can be seen for this compressor the difference between the original and predicted angles does not exceed 0.25 deg for inlet angles and 0.85 deg for outlet angles. It should be noted that the

compressor shown in Figure 10 and Table 3 has averaged geometrical error slightly higher than 10 %. Thus, it can be seen that even at a 10 % averaged geometrical error threshold the discrepancies are not significant. At a 5 % threshold, the quality of prediction of compressor geometry can be considered as high.

Overall the geometrical accuracy of the single AI geometry model for all stages at a 5% error threshold (averaged over all geometrical parameters) is 0.908 for multistage axial compressors from 2 to 15 stages in the validation set.

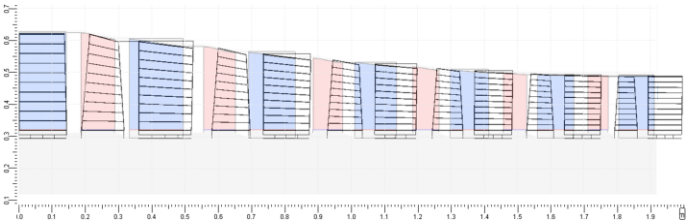


FIGURE 9: COMPRESSOR GEOMETRIES OF DESIGN WITH ID AC118 FROM TEST DATA SET; BLACK OUTLINE: ORIGINAL, COLORED: RECONSTRUCTED

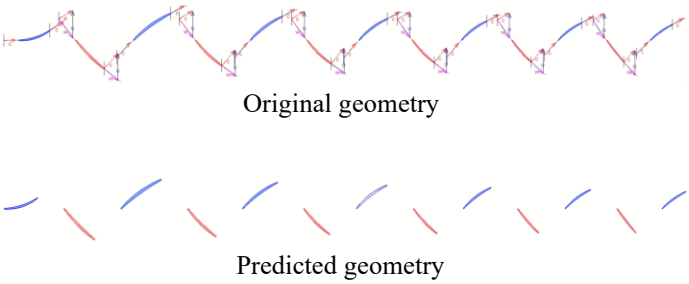


FIGURE 10: BLADE PROFILES AT MEAN SECTION OF DESIGN WITH ID AC966 FROM TEST DATA SET

TABLE 3: INLET AND OUTLET METAL ANGLES AT MEAN SECTION OF DESIGN WITH ID AC966 FROM TEST DATA SET

Row	Inlet Metal Angle			Outlet Metal Angle		
	Orig.	Pred.	Diff.	Orig.	Pred.	Diff.
IGV	90	90	0	54.620	54.620	0
Rotor 1	38.78	38.72	0.06	52.13	52.07	0.06
Stator 1	44.90	44.87	0.03	64.31	64.29	0.02
Rotor 2	37.65	37.55	0.10	55.26	54.94	0.32
Stator 2	42.03	41.98	0.06	65.19	65.19	0.00
Rotor 3	36.80	36.67	0.13	55.73	55.14	0.59
Stator 3	40.76	40.72	0.04	65.61	65.60	0.01
Rotor 4	36.08	35.91	0.17	53.38	52.55	0.83
Stator 4	40.95	41.01	-0.06	65.56	65.52	0.05
Rotor 5	35.45	35.23	0.22	48.69	47.84	0.85
Stator 5	42.57	42.65	-0.08	65.07	65.01	0.06
Rotor 6	34.87	34.61	0.25	42.60	41.93	0.67
Stator 6	45.64	45.77	-0.13	64.17	64.07	0.10

3.2 Hyperparameters Fine-tuning, and Training Results

The process of data sets generation is continuous and started about a year ago. The *Accuracy*$_{5\%}$ of the first models was quite low, e.g. for the first ptr model it was 0.404 and for the eff_tt model, it was 0.688. These results were obtained on the dataset with around 300k performance points. At the moment of writing this paper, the off-design performance dataset included almost 2 million performance points. The performance prediction results presented below are provided for the models trained with the most recent data set.

It should be noted that the most recent validation set consisted of 2625 maps (15 % of the entire data set). In this section, only selected maps will be shown to give readers an understanding of the quality of performance maps predictions along with the uncertainty ranges.

The map #1217 prediction results from the validation set for ptr and eff_tt with quite narrow uncertainty ranges are shown in Figure 11 and Figure 12 respectively. In turn, Figure 13 and Figure 14 show another example (map #195) with pretty wide uncertainty ranges.

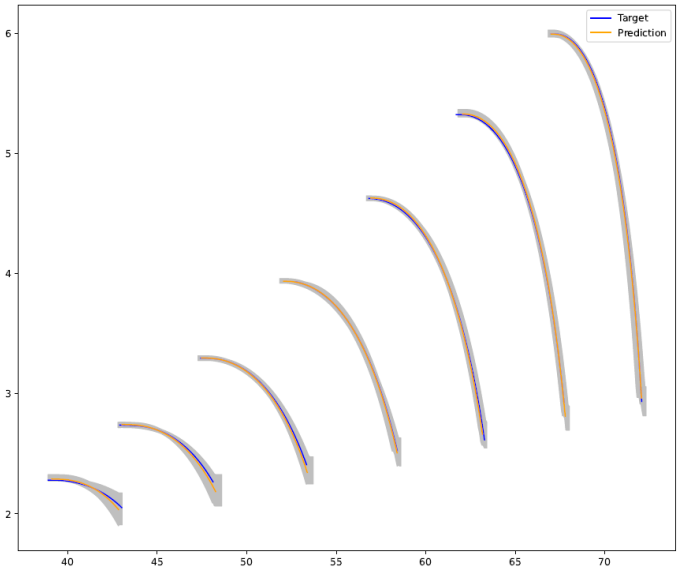


FIGURE 11: COMPRESSOR PERFORMANCE MAP CASE# 1217 FROM VALIDATION SET. X-AXIS IS MFR. Y-AXIS IS PTR. BLUE LINE - VALIDATION SET. ORANGE LINES - PREDICTIONS. GRAY AREAS - UNCERTAINTY RANGES

Grey areas surrounding every speedline on the maps represent the uncertainty ranges. Essentially it means that speedlines predictions of any single model from the ensemble can be anywhere within the grey area. However, averaged prediction among all models in the ensemble eventually gives quite accurate (orange lines) results in both cases even though uncertainty ranges are wider for case #195. This wide uncertainty could be caused by fewer training data points in the vicinity of case #195 in comparison to the case shown in Figure 11, i.e. the model is less confident. Also note, that these are two example

cases out of thousands of maps from the validation set and demonstrate the opposite potential cases, where uncertainties are minimal and maximal. This very well demonstrates the necessity of utilization of ensembles along with AutoKeras to have an accurate prediction of compressor performance maps.

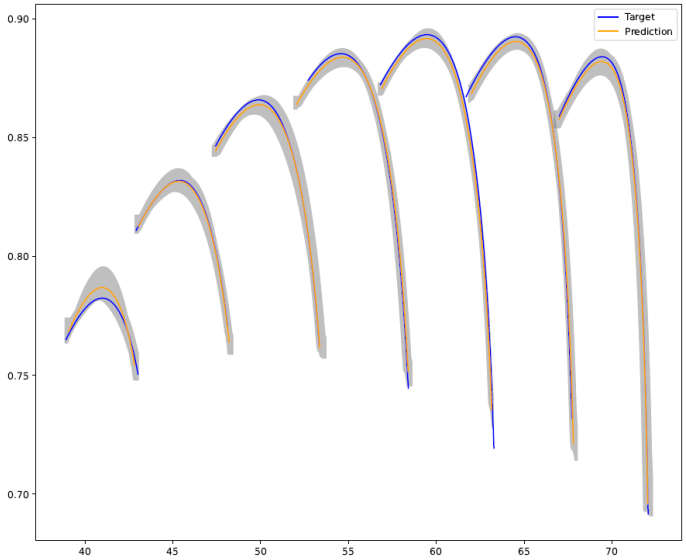


FIGURE 12: COMPRESSOR PERFORMANCE MAP CASE# 1217 FROM VALIDATION SET. X-AXIS IS MFR. Y-AXIS IS EFF_TT. BLUE LINE - VALIDATION SET. ORANGE LINES - PREDICTIONS. GRAY AREAS - UNCERTAINTY RANGES.

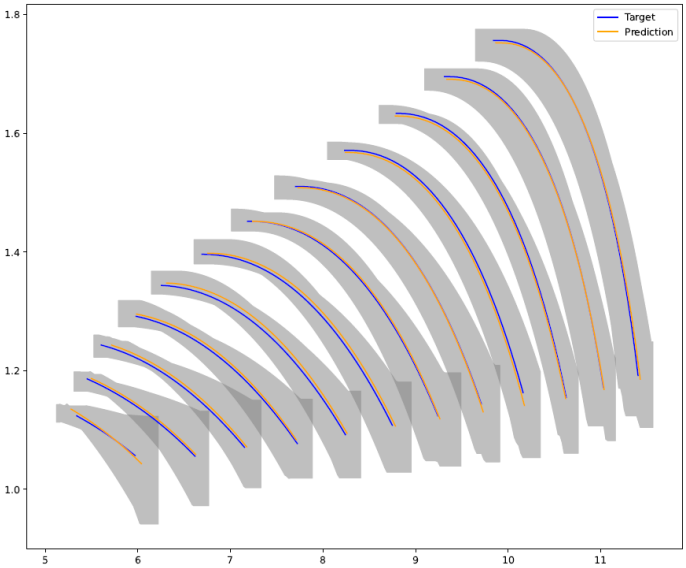


FIGURE 13: COMPRESSOR PERFORMANCE MAP CASE# 195 FROM VALIDATION SET. X-AXIS IS MFR. Y-AXIS IS PTR. BLUE LINE - VALIDATION SET. ORANGE LINES - PREDICTIONS. GRAY AREAS - UNCERTAINTY RANGES.

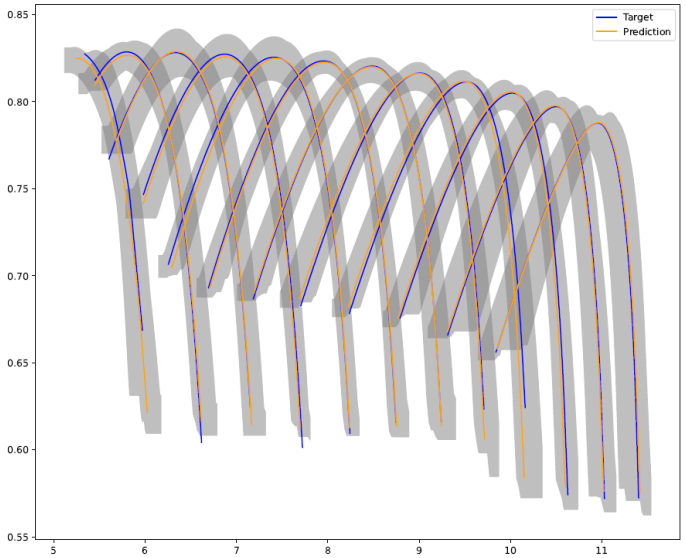


FIGURE 14: COMPRESSOR PERFORMANCE MAP CASE# 195 FROM VALIDATION SET. X-AXIS IS MFR. Y-AXIS IS EFF_TT. BLUE LINE - VALIDATION SET. ORANGE LINES - PREDICTIONS. GRAY AREAS - UNCERTAINTY RANGES.

Overall for the entire validation set $Accuracy_{5\%}$ for the ptr model is 0.978 and for the eff_tt model it is 0.991. The results for psr and eff_ts were not obtained yet at the moment of writing this paper. However, from previous evaluations, it was noticed that usually the accuracy of psr was quite close to the accuracy of ptr and the accuracy of eff_ts was close to eff_tt respectively. The provided above results were for the ensembles with 18 models. The smaller number of models in ensembles was also investigated. Typically the accuracy was less with fewer models in the ensemble. For example, the $Accuracy_{5\%}$ of the ptr model with 6 models in the ensemble was 0.964. It was determined that 18 models are a reasonable trade-off between the accuracy and training time of extra models.

In turn, the investigation of a number of trials in AutoKeras showed that 100 is a Goldilocks value in our case. Going to 150 does not provide substantial improvements in accuracy, but increases the training time substantially. On the other hand, the best model obtained from fewer trials sometimes was not providing the desired accuracy even with multiple uncertainty models.

The training was performed on four workstations with the following specs: 1) CPU AMD Ryzen 9 5950x and GPU NVIDIA RTX3090, 2) Intel i9 12900k and GPU NVIDIA RTX3090, 3) Intel i9 12900k and GPU NVIDIA RTX3090Ti, and 4) 7950x with GPU NVIDIA RTX4090. The amount of RAM was different on different workstations ranging from 64 to 256 Gb, which did not have a sensible impact on the speed of data generation or training. The training was performed using Tensorflow 2.8.0 for GPU. Taking into account that separate models are used for the prediction of ptr and eff_tt it was possible to perform training of the models on different workstations in parallel. On average the training of either the ptr or eff_tt model

was taking from one to two weeks. This is including the 100 AutoKeras trials and training of all uncertainty models. The larger the data set the longer it takes to train the models. The average training time provided above is for the 2 million points data set. It should be noted that the training on RTX4090 was about two times faster than on RTX3090.

Using the trained models the prediction of performance point or entire speedline on average was taking around 0.2 sec on a regular laptop without any inference accelerators. The time for prediction of a performance map is a multiple of the number of desired speedlines, e.g. on average it takes 1 sec to predict a map with 5 speedlines. This is including all uncertainty models and all performance parameters.

3.3 AI Model Utilization in The Cycle Analysis Results

In order to make a time comparison appropriate the cycle described in section 2.4 of this paper was run on the same Intel i9 12900k workstation three times. The first time the original cycle was run without the utilization of any AI models, the second time the cycle calculation was run with the HPC compressor replaced with the AI model and the third time it was run with both LPC and HPC replaced with the AI model.

The time required to run a single part-load point of the engine with the original cycle took around 150 minutes. With the HPC AI, it took around 120 minutes, and with both HPC AI and LPC AI it took around 90 minutes. As can be seen, the replacement of each compressor resulted in about a 20 % time reduction. It should be noted that the cycle still contained the 1D/2D models of Fan, HPT, and LPT which were taking a substantial portion of time in the overall cycle convergence process. It is plausible to assume that with all turbomachinery components replaced with AI models, the time to run the model of the entire turbofan engine could be reduced to a minute or even less, which is comparable to running the simple heat balance calculation of the cycle without any maps or real models of turbomachinery components.

The comparison of the cycle performance values obtained in the original cycle and the cycle with LPC AI and HPC AI is shown in Table 4.

TABLE 4: COMPARISON OF TURBOFAN ENGINE CYCLE CALCULATION RESULTS

Cycle	Original	HPC AI & LPC AI
eff_HPC	0.8808	0.8865
eff_LPC	0.9105	0.9057
eff_HPT	0.8749	0.8749
eff_LPT	0.8849	0.8845
eff_fan	0.8711	0.8711
Thrust, kN	174.133	174.103
Gf, kg/s	1.9801	1.9802
Gfan, kg/s	537.283	538.254

The relative differences in efficiency values of HPC and LPC are 0.6 % and 0.5 % respectively, which are well below the target error threshold of 5% used for the evaluation of the accuracy of the models. Some discrepancy is anticipated because the model used just a limited set of input parameters **Cod** to predict the performance of the unseen before compressors. These compressors were not present in any of the used data sets. In fact, they were designed independently by the other team. Thus, this case served as a verification of the generalization capabilities of the trained models.

To summarize, the replacement of LPC and HPC 1D/2D models with AI models allowed us to reduce the duration of cycle calculation by 40 % without lowering the quality of engine performance determination.

4. CONCLUSION

The presented materials demonstrate the peculiarities of the development of accurate AI models capable of predicting the geometry and performance of multistage axial compressors with variable IGV and multiple VGVs and their utilization in turbofan engine cycle simulation.

The authors proposed a sequential hybrid model approach and flow path parameterization techniques which allowed us to achieve compressor flow path geometry prediction accuracy of 0.908 at a 5 % averaged error threshold for various multistage axial compressors from 2 to 15 stages.

The authors investigated the influence of various hyperparameters of AutoKeras AutoML tool, uncertainty quantification technique, ensembles, and other factors on the accuracy of the compressor performance AI models and achieved the validation set accuracy of 0.978 for the ptr model and 0.991 for eff_tt model at 5 % relative difference (error) threshold by training the ensembles of 18 models for each best model found by AutoML and averaging the predictions.

Using the trained models the prediction of multistage axial compressor performance point or entire speedline on average was taking around 0.2 sec on a regular laptop without any inference accelerators.

The replacement of LPC and HPC 1D/2D models with AI models in the turbofan engine cycle allowed reducing the duration of cycle calculation by 40 % without lowering the quality of engine performance determination, which can substantially reduce the cost and development time of gas turbine engines.

The developed technology provides the solid foundation for the development of AI models for other turbomachinery types as well as other components of gas turbine engines and power plants.

ACKNOWLEDGEMENTS

We wish to thank the many people from the SoftInWay, Inc. team who generously contributed their time and effort in the preparation of this work.

The information, data, or work presented herein was funded in part by the SBIR Phase II, NASA Shared Services Center (NSSC), under Award Number **80NSSC21C0558**. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

REFERENCES

[1] Burlaka, Maksym. Moroz, Leonid. "Axial Compressor Map Generation Leveraging Autonomous Self-Training Artificial Intelligence" *J. Eng. Gas Turbines and Power.* Jan 2023, 145(1): 011001 (11 pages): DOI 10.1115/1.4055633

[2] Hongsheng, Jiang. Dong, Sujun. Zheng, Liu. Yue, He. and Fengming, Ai. "Performance Prediction of the Centrifugal Compressor Based on a Limited Number of Sample Data." *Mathematical Problems in Engineering* Vol. 2019 (2019): Article ID 5954128, DOI 10.1155/2019/5954128

[3] Gholamrezaei, Mohammad. and Ghorbanian, Kaveh. "Compressor Map Generation Using a Feed-Forward Neural Network and Rig Data." *Proceedings of the Institution of Mechanical Engineers Part A Journal of Power and Energy* 224(1): pp. 97-108. Tehran, Iran, February 2010. DOI 10.1243/09576509JPE792

[4] Fei, Jingzhou. Zhao, Ningbo. Shi, Yong. Feng, Yongming. and Wang, Zhongwei. "Compressor Performance Prediction Using a Novel Feed-forward Neural Network Based on Gaussian Kernel." *Advances in Mechanical Engineering* Vol. 8 (2016): DOI 10.1177/1687814016628396

[5] Azzam, Mark; Haag, Jan-Christoph; Jeschke, Peter. Application concept of artificial neural networks for turbomachinery design. Computer Assisted Methods in Engineering and Science, [S.l.], v. 16, n. 2, p. 143-160, Jan. 2017. ISSN 2299-3649. Available at: https://cames.ippt.pan.pl/index.php/cames/article/view/158 [Accessed: 10-Feb-2023]

[6] Moroz, Leonid. Govoruschenko, Yuri. and Pagur, Petr. "A Uniform Approach to Conceptual Design of Axial Turbine / Compressor Flow Path" *Proceedings of the Future of Gas Turbine Technology 3rd International Conference* (2006): Brussels, Belgium, October 2006

[7] Wikipedia.org. "ISO 10303-21" Available: https://en.wikipedia.org/wiki/ISO_10303-21 [Accessed: 2-Jan-2023]

[8] Wikipedia.org. "CGNS" Available: https://en.wikipedia.org/wiki/CGNS [Accessed: 2-Jan-2023]

[9] Wikipedia.org. "Compressor map" Available: https://en.wikipedia.org/wiki/Compressor_map [Accessed: 3-Jan-2023]

[10] Sobol, I.M. (1967), "Distribution of points in a cube and approximate evaluation of integrals". Zh. Vych. Mat. Mat. Fiz. 7: 784–802 (in Russian); U.S.S.R Comput. Maths. Math. Phys. 7: 86–112 (in English)

[11] Wikipedia.org. "Bezier curve" Available: https://en.wikipedia.org/wiki/B%C3%A9zier_curve [Accessed: 4-Jan-2023]

[12] Wikipedia.org. "Composite Bezier curve." Available: Composite Bézier curve - Wikipedia [Accessed: 10- Feb-2023]

[13] Qingquan, Song. Haifeng, Jin. And Xia, Hu. "Automated Machine Learning in Action", March 2022, ISBN 9781617298059 – 336 pages

[14] Qingquan, Song. Haifeng, Jin. And Xia, Hu. "Auto-Keras: An Efficient Neural Architecture Search System", Department of Computer Science and Engineering, Texas A&M University, arXiv:1806.10282v3

[15] Haifeng, Jin. "Efficient neural architecture search for automated deep learning" Dissertation submitted to the Office of Graduate and Professional Studies of Texas A&M University, 2021

[16] Ferreira, Luis; Pilastri, Andre; Martins, Carlos; Pires, Pedro; and Cortez, Paulo. "A Comparison of AutoML Tools for Machine Learning, Deep Learning, and XGBoost". Available at: https://repositorium.sdum.uminho.pt/bitstream/1822/74125/1/automl_ijcnn.pdf [Accessed: 15-Feb-2023]

[17] Auto-Gluon, "AutoGluon: AutoML Toolkit for Deep Learning — AutoGluon Documentation 0.0.1 documentation," 2020. Available at: https://autogluon.mxnet.io/ [Accessed: 15-Feb-2023]

[18] D. Cook, Practical machine learning with H2O: powerful, scalable techniques for deep learning and AI. " O'Reilly Media, Inc.", 2016.

[19] Keras.io. "Layer weight initializers" Available: https://keras.io/api/layers/initializers/ [Accessed: 6-Jan-2023]

[20] Xavier, Glorot. Yoshua, Bengio. "Understanding the difficulty of training deep feedforward neural networks" *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics,* PMLR 9:249-256, 2010

[21] Kaiming, He. Xiangyu, Zhang. Shaoqing, Ren. and Jia, Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification" arXiv:1502.01852

[22] Wikipedia.org. "Autoencoder" Available: https://en.wikipedia.org/wiki/Autoencoder [Accessed: 5-Jan-2023]

[23] Clement, Joly. and other. "Digital Twin Application to Enhance Turbofan Engine Performance" *Turbine Engine Technology Symposium 2022,* Dayton Convention Center, Dayton, Ohio, September 12-15, 2022