Proceedings of the ASME 2021 Turbomachinery Technical Conference & Exposition
Turbo Expo 2021
June 7-11, 2021, Virtual, Online

GT2021-1104-60158

USING AUTOENCODERS AND OUTPUT CONSOLIDATION TO IMPROVE MACHINE LEARNING MODELS FOR TURBOMACHINERY APPLICATIONS

Julie Pongetti[*]
Timoleon Kipouros
Engineering Department
University of Cambridge
Cambridge, CB2 1PZ
UK

Marc Emmanuelli
Richard Ahlfeld
Monolith AI
12-18 Hoxton Street
London, N1 6NG
UK

Shahrokh Shahpar
Innovation Hub
Future Methods
Rolls-Royce
Derby, DE24 8BJ
UK

ABSTRACT

Machine learning models are becoming an increasingly popular way to exploit data from fluid dynamics simulations. This project investigates how autoencoders and output consolidation can be used to increase the accuracy of machine learning models, by injecting knowledge of the full flow field in the predictions of Quantities of Interest (QoI) used in the optimisation of highly loaded transonic compressor blades. Results show that the accuracy of the predicted QoI can indeed be increased, by using both an appropriate autoencoder and output consolidation. Most significantly, the prediction accuracy is increased in the range of QoI values which is involved in optimisation problems. As a result, a more accurate and faster computational design approach driven by machine learning methods has been demonstrated.

INTRODUCTION

High-resolution flow simulations are essential design tools in fields such as turbomachinery. It is necessary to accurately predict the performance of a part early in its design phase to reduce costs and time taken, but high-resolution flow simulations are also computationally expensive (and hence costly) to use. This makes it prohibitive to run large numbers of simulations in order to optimise a design, with potential losses in terms of final component performance.

Deep Learning techniques are now powerful enough to extract knowledge from data that has so far been too complex to analyse. Machine learning can therefore be a revolutionary tool, by using existing simulation data (which would otherwise become redundant under any minor design change) to predict the influence of various parameters on quantities significant for optimisation. A part can therefore be optimised in a much shorter time, by exploring new designs in real time and with the possibility of automating all the steps in the design cycle, greatly reducing the design costs.

For these reasons, the use of machine learning techniques for fluid dynamics applications has recently attracted attention. An extensive description of how fluid mechanics problems can be set as machine learning tasks is given in [1]. A wide range of studies can also be found in the literature, including some on turbomachinery design and optimisation using Neural Networks [2] or multi-fidelity Gaussian Processes [3]. In other applications there are already examples of real-time 3-dimensional flow predictions which allow interactive design of complex shapes [4]. These more advanced interactive optimisation examples both exploit the information contained in the CAD designs by pre-processing the 3D data and using it to predict the flow field.

In this paper we explore how, when a limited dataset is available, the combined use of autoencoders and output consolidation can be used to improve the accuracy of machine learning models and allow optimisation of turbomachinery components.

[*] Address all correspondence to this author. Email: jp772@cam.ac.uk

MACHINE LEARNING IN TURBOMACHINERY

Computational design processes in turbomachinery for the development of components and subsystems typically require advanced CFD simulations in order to enable an optimization algorithm to explore the design space effectively. This is an inherently computationally expensive process and despite the continuous advancements of PC hardware and optimization algorithms, the need for increased intelligence in the process is apparent. Machine Learning, Response Surface Methods, and Surrogate Models have been studied and integrated in these processes for long time [5–8]. There are a number of ways to aid the design process using Machine Learning models and different strategies [9–11]. Each method offers its own advantages, but often compromises on other aspects of the process, or even on the understanding and interpretation of the improved designs.

Looking then at the most recent developments, researchers [12] have developed an accurate turbulence modelling closure for wake mixing prediction of a turbine blade using gene expression modelling (or what used to be called *genetic-programming*). By first using a canonical turbine wake with inlet conditions prescribed based on high-fidelity data, it was demonstrated that the RANS-based CFD-driven machine-learning approach produces non-linear turbulence closures that are physically correct, i.e. predict the right downstream wake development and maintain an accurate peak wake loss throughout the domain. However, it was also found that the model was too diffusive for URANS calculations, interestingly demonstrating that the model accuracy depends on whether it will be deployed in RANS or unsteady RANS calculations.

The application of gene expression programming was also extended to augment RANS turbulence closure modelling for flows through complex geometry, designed for additive manufacturing. Specifically, for the design of optimised internal cooling channels in turbine blades [13].

Researchers in [14] develop a bespoke heat-flux closure using the machine-learning approach for the stochastic heat-flux of a canonical flow of pressure-side bleed flow encountered in high pressure turbine TE slots. The machine-learnt closures developed specifically for URANS calculations show significantly improved predictions for the adiabatic wall-effectiveness across the different cases.

Although the generality of these tools comes at the cost of predictive power, it is shown [15] that this can be recovered using supervised machine learning. Supervised machine learning can be leveraged to improve online prediction of thermos-acoustic combustion instabilities using dynamic pressure readings [15].

PROBLEM UNDER STUDY

NASA Rotor 37

The transonic compressor blade under study is the NASA Rotor 37. This is a highly loaded blade developed at the NASA Lewis Research Center which exhibits large shock-induced separation and complex flow behaviour. NASA Rotor 37 is a well-documented case, having been extensively tested [16, 17] and simulated as part of a turbomachinery validation study [16–19] and used in multi-disciplinary optimisation studies [20] using trust-based DOE+RSM methods. The very high pressure ratio, strong shock wave boundary layer interaction, large tip-leakage vortex and highly separated flow mean that it poses challenges for turbomachinery solvers and has been the subject of review articles that highlight the complexity of matching experimental and computational measurements and the associated uncertainties [21,22]. The CFD setup of the simulations used in this study is shown in Fig.1. At the inlet, a radial distribution of total pressure and temperature (based on the original experimental values [18]) is specified. At the outlet, the value for circumferentially mixed-out and radially mass-meaned capacity ($\dot{m}\sqrt{T_{exit}}/p_{exit}$) is specified to ensure comparability of the results. Stationary walls are treated as adiabatic viscous walls and the rotational speed is 1800.01 rad/sec. Rolls-Royce CFD solver Hydra is used for all of the simulations presented here, using the Spalart-Allmaras turbulence model. For a validation of Rotor 37 simulations in Hydra and a discussion on uncertainties, refer to [23]. The mesh independent grids with y+ 1 used are generated by PADRAM [24,25].

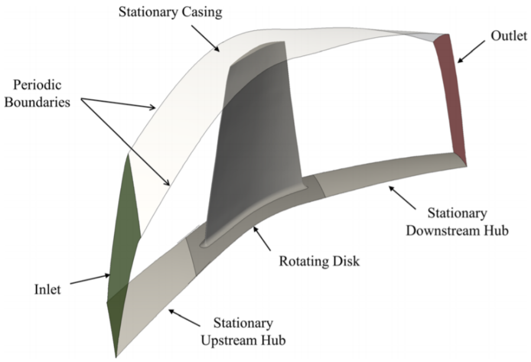

FIGURE 1. DOMAIN FOR CFD SIMULATIONS.

Design space

The PADRAM so-called Engineering Design Parameters (EDPs) are used here. These deformation modes are listed below and can be seen in Fig.2:

1. *Skew*, which is responsible for the solid body rotation of sections of the blade;
2. *Delt* (or *tangential lean*), which is responsible for the circumferential movement of sections leading to lean of the blade;
3. *Xcen* (or *axial sweep*), which is responsible for the axial

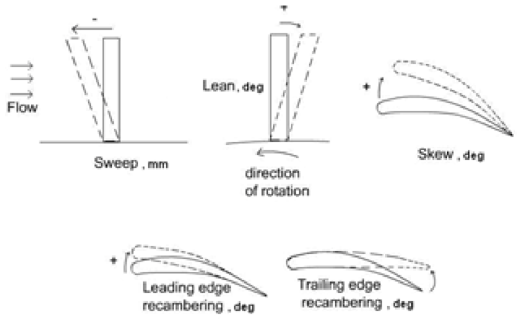

FIGURE 2. REPRESENTATION OF THE DEFORMATION MODES.

movement of sections along the engine axis leading to a sweep effect;

4. *Lemo*, which is responsible for the leading-edge (LE) recambering;
5. *Temo*, which is responsible for the trailing-edge (TE) recambering.

There are two additional EDPs that control the locality of the LE and TE recambering. This is measured on the %chord where the perturbation starts. For LE recambering, values of 0.1 imply very localized perturbations, while values of 0.9 imply global modification, and vice-versa for the TE recambering.

The values for these seven parameters are specified at 5 uniformly distributed span-wise locations (0%, 25%, 50%, 75%, 100%), yielding a total of 35 design variables. The resulting deformation of the whole 3D surface is derived from a B-Spline interpolation where each aerofoil section serves as a knot.

Objective function

The Quantitity of Interest, or objective function, in this study is the adiabatic efficiency (η) as given below:

$$\eta = \frac{(P_{outlet}/P_{inlet})^{(\gamma-1)/\gamma}-1}{T_{outlet}/T_{inlet}-1} \qquad (1)$$

Here P and T are the total mass averaged pressure and temperature, respectively. Constraints are placed on the pressure ratio and mass flow rate, which have to be within a set range of the characteristic values for the datum blade, i.e. 2.08 ± 1.0% for pressure ratio and 20.1 kg/s ± 0.5% for mass flow rate. The aim is to maximise the adiabatic efficiency under these constraint by varying the geometry of the blade.

Train and test datasets

The results of 390 CFD simulations are available for this study. More specifically, four types of data are taken from each simulation: tabulated values of the EPDs and the resulting QoI; 3D surface mesh of the blade; tabulated blade surface pressure field (XYZ coordinates of each mesh point and corresponding value of pressure); tabulated exit total pressure (P_0) and exit total temperature (T_0) fields (XYZ coordinates of each point and corresponding values of stagnation quantities).

The samples were generated using an LPTau DOE, a pseudo-random method which uses Sobol sequences, that usually provides a good coverage of the design space. Specifically, in the present case which involves a reduced dataset, this method was considered superior to other randomized sampling techniques such as Latin hypercube, which would have a stronger bias towards the corners of the design space, or Monte Carlo (with uniform probability density functions), which would not ensure a sufficient spread.

A rule of thumb (10×(design parameters) + 50) was followed to define the size of the overall dataset. For the purpose of training and evaluating the performance of the machine learning models, the available simulations are split in a train data set (350) and a test data set (40). While larger testing sets are usually preferred to assess how a model generalizes to unseen data, in this case increasing the size beyond 10% would have limited the amount of training data to a possibly insufficient number of samples, with consequences on the ability to derive relations.

As autoencoders (described below) work only on meshes with the same structure (number of mesh points), this reduces the number of simulations available for most parts of this study. For models which take as inputs latent parameters extracted by the autoencoder, or which require predictions obtained from those, the available training and test sets are reduced to 301 and 35 simulations respectively. This is due to the fact that PADRAM always generates a new surface mesh from the blueprint described by the vector of EDPs. This makes it very fast to produce new geometries, but as the different surface meshes are not obtained by morphing one single original design, additional nodes may be placed if necessary to ensure good quality. This causes some of the geometries for which CFD simulations are available to be unsuitable for use with the autoencoder.

A further set of 4000 surface meshes covering the design space of the EDPs is then provided specifically for the training of the autoencoder. Note that unlike running CFD simulations for the flow, generating surface meshes for the blades using PADRAM is a fast and inexpensive process.

TOOLS AVAILABLE

Before the proposed methodology can be described, an outline of the software capabilities and machine learning techniques available is presented.

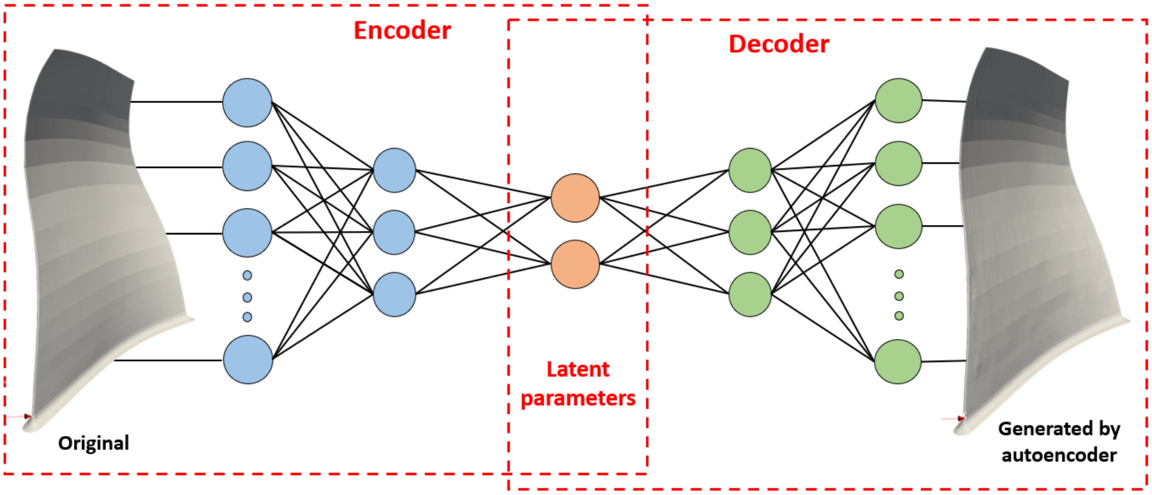

FIGURE 3. STRUCTURE OF AN AUTOENCODER: THE INPUT GEOMETRY IS CONDENSED IN A SET OF LATENT PARAMETERS (ENCODER); THE LATENT PARAMETERS ARE USED TO GENERATE AN NEW GEOMETRY (DECODER). NOTE THAT THE AUTOENCODER READS AND OUTPUTS THE COORDINATES OF THE MESH POINTS OF THE BLADES.

Machine learning framework

This study was carried out using Monolith[1], a machine learning software specifically conceived for engineering applications.

The models created in this paper are implemented through the software using standard machine learning frameworks: the autoencoder and Neural Networks use the TensorFlow library [26], whereas Gaussian Process Regressions use the Sklearn library [27]. An outline of these is given below.

The hyperparameters defining the NNs are (default value in parentheses): n_epochs (100), batch_size (32), hidden_layer_sizes (100,100), dropout_probability (0.05), activation_function ("relu"), scoring metric (mean squared error). Details about the meaning of each hyperparameter can be found by consulting TensorFlow documentation.

Different covariance functions (kernels) are available for the GPR: Matern1/2, Matern3/2, Matern5/2, RBF, and WhiteKernel. This last one, also referred to as a Noise kernel, explains the noise component between input and outputs, whereas the other four kernels determine the smoothness of the resulting functional relationship (respectively: not differentiable, differentiable once, twice, or infinite times). Refer to [28] for a mathematical formulation of the different kernels in the context of Gaussian Processes in machine learning.

Autoencoders

Variational autoencoders are a specific type of Neural Network, which allow to feed non-parametric CAD designs into machine learning models. The software offers an autoencoder model which can be tuned by the user and which is designed to extract a (user-specified) number of latent parameters describing the given 3D geometry. Once generated, these latent parameters can be used as an alternative to the EDPs in describing the shape of each surface mesh: the parameters make up a vector of specified length, where the value of each entry describes a feature of the geometry. For a trained autoencoder it is possible to investigate what each latent parameter represents by observing how varying the value of the specific entry changes the shape of blade. In general, autoencoders can be a powerful tool because the parametrisation found by the machine learning model might offer a more thorough or insightful way of describing the geometry. For example, a latent parameter might describe a combination of lean and sweep of the blade, in a way that is more intuitive or that better captures a specific impact on the QoI.

The autoencoder is defined through a series of hyperparameters: number of latent parameters, complexity, batch size, and training epochs. The last two parameters specify how the training of the model is carried out. The complexity instead determines the internal structure of the encoder and decoder: geometries which include a large number of features will require a greater number of hidden neurons to model it satisfactorily. The complexity is indicated on a scale from 1 to 10 and its value can initially be estimated from the geometry. For example, square boxes would have a complexity of 1, whereas turbine blades with cooling holes would likely be represented by a value of 10.

The autoencoder is first trained on a set of surface meshes with the same structure to identify the latent parameters which describe the object. The autoencoder takes as inputs an STL file,

[1] More information is available at: `http://www.monolithai.com`

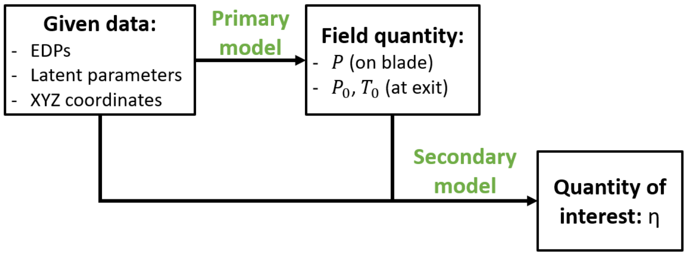

FIGURE 4. CONNECTION BETWEEN MODELS IN OUTPUT CONSOLIDATION.

which must have a fixed number of vertices (in this study 11162, which was the number of points present in the majority of the PADRAM generated meshes). Once training is completed, two models are available: an encoder and a decoder (see Fig.3). From a 3D surface mesh, the encoder can obtain the values of the latent parameters describing that specific geometry; from chosen values of the latent parameters, the decoder can generate the corresponding 3D mesh. It is important to note that the autoencoder does not require any data from expensive simulations, but is based simply on surface mesh files which can be easily generated in large numbers.

Output consolidation

Quantities of Interest (QoI) describing the performance of a turbomachine are normally calculated from relevant outputs of flow simulations, such as pressure fields or velocity profiles, as well as integral values like the mass-averaged total pressure, total temperature and swirl at inlet and exit. Output consolidation is a method which allows to incorporate multiple relevant data sets into a single specific prediction. It uses two models in succession (see Fig.4), in a way such that:

– The primary model predicts an intermediate quantity (related to the QoI) from the given scalar or geometry data;
– The secondary model predicts the QoI by taking as inputs both the given data and the primary model predictions.

Modelling the intermediate quantities which are used to calculate the QoI is a way of incorporating more knowledge in the final machine learning model, which becomes informed about the flow.

PROPOSED METHODOLOGY

Neural Networks or Regression methods often take a set of inputs parameters (which represent the geometry, boundary conditions, etc.) and predict outputs such as lift coefficient or efficiency. This limits the possible applications, both by excluding the use of non-parametric designs and by restricting the outputs to a few significant quantities. In this paper we explore for the first time the use of autoencoders and output consolidation as a novel way to mitigate these restrictions and obtain increasingly accurate predictions.

Baseline models

Baseline models are first developed using standard techniques to establish a ground for comparison of subsequent models.

Gaussian Process Regressions (GPR) and Neural Networks (NN) taking as inputs different subsets of EDPs are considered as possible models and their performance is evaluated.

For the GPR, kernels are tried independently or by pairing the best performing one with some of the others. Models using all five kernels at once are also considered.

When NNs are used, the number of epochs is adjusted based on the training fitness convergence (evolution of the loss function during training). A first value for batch size is chosen depending on the size of the training set, and then this is varied by grid search along with the hidden layer size (more specifically, with the number of neurons per layer). Three to four different values are tried for each quantity to find the model structure which leads to the best prediction accuracy.

One simulation is removed from the train set due to the noticeably low value for the QoI. Discarding this outlier from the otherwise quite uniform distribution of the train set improves the performance. Correlations between the 35 parametric variables and QoI are investigated, and input reduction based on this analysis leads to significant improvements for this specific problem.

Two baseline models are developed to provide different assessment criteria: one model is trained on all the available simulations (349, after the removal of the outlier); the other model is trained only on the simulations (301) whose geometry is compatible with the autoencoder. These two models will allow us to draw different conclusions: with the former model it will be possible to evaluate how many high cost CFD simulations can be spared using the presented method; on the other side, we will establish by how much the accuracy of a model can be increased through our approach.

Development of the autoencoder

The first part of the work focuses on training an autoencoder to extract latent parameters from the geometry. These parameters can then be used to complement or replace traditional scalar values defining the geometry, and to predict complex 3D fields through deep learning. In this study, latent parameters are seen to offer better field predictions than what is achievable with the EDPs. A significant advantage of autoencoders is that they offer the possibility of visualising decoded geometries in real time within the software. This can be more intuitive than reading through tables of scalar parameters and facilitates quick comparison of designs without the need of exporting the results to

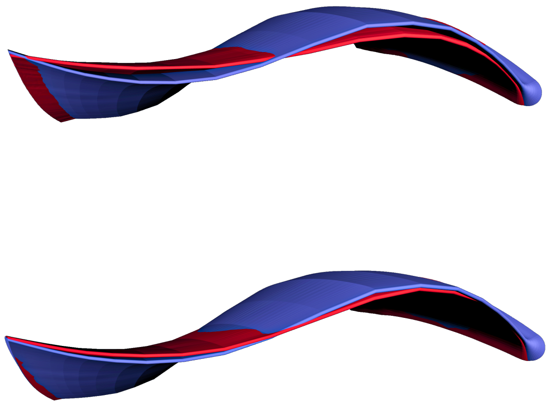

FIGURE 5. EXAMPLE OF ORIGINAL (BLUE) AND DECODED (RED) GEOMETRIES FROM THE TEST SET; TOP: AUTOENCODER TRAINED ON 301 MESHES, BOTTOM: AUTOENCODER TRAINED ON 4000 MESHES.

external applications.

Autoencoders with different hyperparameters are trained on a set of meshes and then compared by assessing the reconstruction accuracy for a set of unseen meshes. The software allows to visualise superposed the original (test) geometry and the reconstructed one, which makes it possible to visually identify the autoencoder with the best accuracy (Fig.5).

The autoencoder was initially trained on the geometries of the 301 available CFD simulations, but a more thorough set of 4000 surface meshes spanning the design space was later generated specifically for this purpose. Due to the simplicity of defining blades in PADRAM, this improved the accuracy of the trained autoencoder, while incurring little cost in terms of time or complexity.

The extracted latent parameters are used in the subsequent models to integrate and augment the EDPs defining the geometry.

Approach to output consolidation

Output consolidation is carried out using two sets of simulation results:

1. Surface pressure field on the mid-span section of the blade;
2. Exit stagnation pressure and temperature on a selected downstream plane.

Output consolidation is applied to each dataset independently, therefore two alternative predictions of the QoI are obtained and can be compared.

Primary and secondary models are developed for each dataset. The primary model is in all cases a Neural Network, as the quantity to be modelled is a field and the large amount of training data cannot be handled by Regression models. Similarly to what was done for the baseline case, both GPRs and NNs are trained as a secondary model; when the former method is used, training is based only on a random subset of the field predicted by the primary model. Random subsets of size ranging from 1,000 to 4,000 data points are compared, obtaining the best performance from GPRs trained on about 2,000 data points (or around 0.6% of the over 300,000 original field points).

Again, for each situation different kernels or hyperparameters combinations are compared (see subsection “Baseline models”). The best model in each case is selected based on the performance on the unseen test dataset.

When developing the models for consolidation, a major role is played by the choice of inputs to each model. The secondary model necessarily takes predictions from the primary model, but these are only meant to help “refine” predictions which are mainly based on the key geometry quantities. As for the primary model, the accuracy of its predictions are dependent on a proper description of the geometry. For this reason, a significant portion of the work on consolidation involved analysing the results obtainable for different combinations of input variables.

Models for optimisation

As detailed in the results section, the most accurate secondary models only take as geometric inputs a subset of the parametric variables. If optimisation were to be carried out using these models, the pruned inputs would then need to be recovered by other methods so as to obtain a complete description of the blade geometry. Here we develop instead a final (“tertiary”) model which takes all the geometric parameters as inputs, and which recovers all values directly upon optimisation.

Two variants are trained: both take as inputs the predictions of the QoI obtained from the secondary models, but the geometry is defined differently in the two cases. One model uses the vector of 35 EDPs, whereas the other takes all the latent parameters extracted by the autoencoder. These tertiary models, which also have the QoI as an output, can then be used for optimisation, returning complete sets of parameters describing the blade geometry.

The functional relationship for the final models is expected to have the same features as that of the baseline models, therefore the same type of model (a GPR, with the same kernels) is chosen.

A targeted optimisation (based on differential evolution) without any geometry constraints is run on the two tertiary models to maximise the QoI. Given a number of candidates in the design space, the evolutionary algorithm finds the optimal solution by iteratively combining attributes of the best candidates found in the previous step.

The 3D surface meshes for the two models are then generated from the EDPs or latent parameters and CFD simulations are run to evaluate the actual performance of the optimised designs.

Assessment of model accuracy and compared performance

For all models, the accuracy on the unseen test data is calculated and the performance is compared based on 4 parameters:

1. Mean Absolute Error (MAE): the absolute value of the difference between the predicted and actual values; has a low value in accurate models.
2. Mean Squared Error (MSE): this is calculated as the average squared difference between the predicted and actual values; a value closer to zero denotes a more accurate model.
3. Pearson Coefficient: measures the strength of the association between the predicted and actual values; a value of +1 is the result of a perfect positive relationship, so values closer to unity are desirable.
4. R-squared value: measures the proportion of the variance of the predicted quantity that is explained by the variation of input quantities; a value of 1 means that all the variation of the predicted value is explained by variations in the input quantities, therefore values close to unity indicate a good model.

The last two quantities are generally considered more significant for the predictions of the QoI, as they give an indication of how well the model is capturing trends in the data. The MSE will be used in preference when the accuracy of a specific restricted range of values is being examined, whereas the MAE is used when assessing predictions of the field quantities.

RESULTS

The details and accuracy of the best-performing models are presented below.

Selected baseline models

Gaussian Process Regressions were generally seen to perform better for what concerns baseline predictions. More specifically, the combined use of kernel *Matern1/2* and *Noise* showed the best results. Note that the models taking all 35 EDPs as inputs performed poorly due to the highly non-linear nature of the problem and limited training data available.

To mitigate this problem, Linear, Power Law, and Exponential fits were obtained between each parametric variable and the QoI, with the strongest relationship being retained for each pair. Only the variables showing a higher correlation with the QoI were then used as inputs to the models, removing part of the noise and enabling more accurate predictions. Based on the correlation strength, different combinations of inputs were compared to find the one yielding the best accuracy on the test set.

More specifically, the 13 EDPs that play a key role in QoI predictions and which are widely used in the models developed in this study, are: sweep (100%), lean (0%, 25%, 100%), skew (100%), LE recambering (0% to 75%), TE recambering (25%, 50%), locality of LE recambering (0%), locality of TE recambering (100%). Note that this does not make the other EDPs redundant, as their value still affects the performance of the blade; the EDPs selected here are those which allow the most accurate predictions to be obtained from the models, given the limited training data available.

The chosen models “Baseline (301)” and “Baseline (349)” have structure:

- Model: GPR with kernels *Matern1/2* and *Noise*;
- Inputs: the 13 EDPs listed above;
- Output: QoI.

Performance of autoencoder

The performance of the initial autoencoder trained on the 301 compatible surface meshes available in the training set was not deemed to be fully satisfactory. A separate set of 4000 surface meshes spanning the design space of the EDPs were later generated, which resulted in a more accurate autoencoder that better captured the shape of the blades (see Fig.5).

The selected autoencoder which is used to parametrise the geometries for the subsequent work has the following hyperparameters: number of latent parameters = 30; batch size = 10; complexity = 8 (out of 10); training epochs = 200,000[2].

A high number of training epochs was chosen to ensure convergence of the model; low batch size and higher number of latent parameters (30 is the maximum number offered by the software) were seen to improve accuracy. The complexity was selected by trial-and-error, comparing the accuracy of reconstructed (test set) geometries.

Consolidation with blade pressure field

Primary model. The primary model performed much better when based on latent parameters only rather than on the EDPs exclusively. However, the best accuracy was obtained when the use of latent parameters was complemented with the EDPs describing the shape of the central span of the blade. Physically, reinforcing the geometric description of the concerned section of the blade can be interpreted as an advantage for the model.

The primary model used for this part has the following characteristics:

- Model: Neural Network (epochs = 150, batch size = 120, neurons per hidden layer = 150);

[2] The research was hosted on a cloud-based high-performance computing service (Amazon Elastic Compute Cloud), using central processing units (CPU). Here, the training time was 30 minutes for 200,000 epochs.

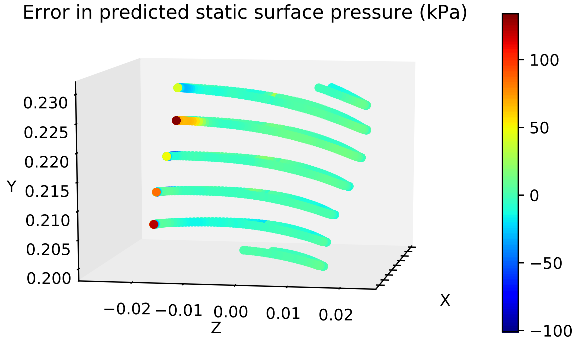

FIGURE 6. ERROR IN PREDICTED STATIC PRESSURE FIELD ON THE PRESSURE SURFACE OF A BLADE; COLOURED BY STATIC PRESSURE (kPa).

- Inputs: XYZ coordinates of the mesh point, all latent parameters, 15 parametric variables describing the central section of the blade (25% to 75% span);
- Output: static pressure.

The accuracy of the model can be assessed by looking at the MAE of the predictions for the test set. In this case, the MAE of 7.43kPa is about 6% of the mean pressure value on the surface of all blades, which is 119.12kPa. Note that the pressure shows large variations across the surface of the blades in the test set, with a minimum value of 17.42kPa and a maximum of 244.97kPa.

A further improvement for this model could possibly be achieved by predicting the pressure values on the pressure and suction surfaces of the blade separately, as results show that the highest error occurs near the leading edge of the blade. This can be seen in Fig.6, which compares the actual and predicted static pressure on the pressure surface of the blade: the flow goes from left to right in both cases, with the LE predictions having the greatest error.

Secondary model. The selected secondary model ("Consolidated blade") has a structure similar to that of the baseline models and uses the same EDPs to describe the geometry:

- Model: GPR with kernels *Matern1/2* and *Noise*;
- Inputs: XYZ coordinates of the mesh point and corresponding predicted pressure (from primary), 13 EDPs (as in the baseline models);
- Output: QoI.

Comparisons of the Pearson Coefficient and R-squared value for this secondary and the baseline models is shown in Figure 7. Compared to the baseline models, the consolidated model achieves:

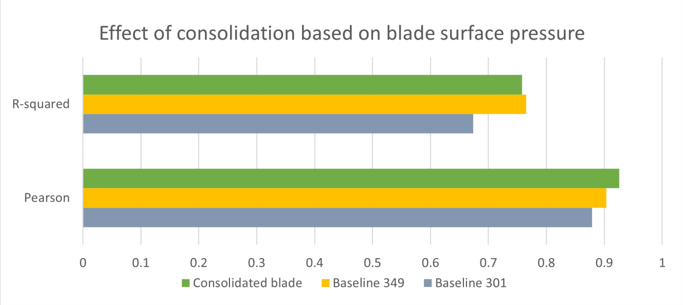

FIGURE 7. OVERALL PERFORMANCE OF OUTPUT CONSOLIDATION (FROM SURFACE STATIC PRESSURE).

- Overall 5% higher Pearson Coefficient and 12% higher R-squared than a standard model trained on the same number of simulations;
- Comparable overall performance to a standard model trained on 14% fewer simulations;
- For QoI values above 83.5, the MSE is reduced by 67% (from Baseline 301) or 49% (from Baseline 349).

Consolidation with exit stagnation quantities

Primary model. Similarly to what was observed when modelling the blade surface static pressure, the primary model predicting the exit plane conditions performed better when it included latent parameters. In this case, the best results were obtained with the addition of the 13 EDPs used in the baseline models.

The primary model chosen for this part has the following characteristics:

- Model: Neural Network (epochs = 100, batch size = 100, neurons per hidden layer = 100);
- Inputs: YZ coordinates of the point (the exit plane is at constant X), all latent parameters, 13 EDPs (as in the baseline models);
- Output: exit stagnation pressure, exit stagnation temperature.

Predictions of the test set fields for both quantities were more accurate than seen for the blade surface pressure (Fig. 8). More specifically: for the total pressure MAE = 6.33kPa, which corresponds to 3.5% of the mean value; for the total temperature MAE = 2.5K, which is less than 1% of the mean value. The two quantities also showed a smaller range across the test set: 114.01kPa $\leq P_0 \leq$ 259.37kPa and 351.8K $\leq T_0 \leq$ 406.6K.

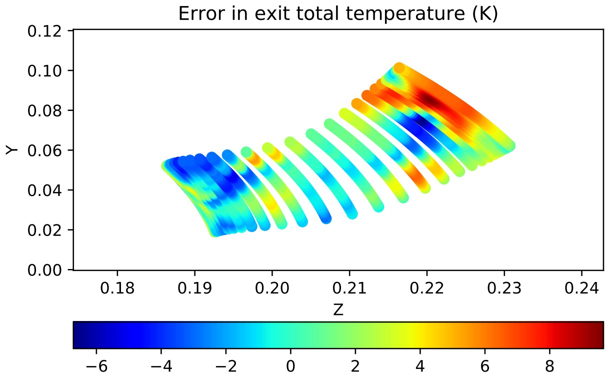

FIGURE 8. ERROR IN PREDICTED TOTAL TEMPERATURE FIELD AT THE EXIT PLANE OF A BLADE. COLOURED BY TOTAL TEMPERATURE.

Secondary model. The selected secondary model ("Consolidated exit") follows the same structure as the other models predicting the QoI:

- Model: GPR with kernels *Matern1/2* and *Noise*;
- Inputs: YZ coordinates of the mesh point and corresponding predicted exit total temperature and pressure (from primary), 13 EDPs (as in the baseline models);
- Output: QoI.

Comparisons of the Pearson Coefficient and R-squared value for this secondary and the baseline models is shown in Figure 9. Compared to the baseline models, the consolidated model achieves:

- Overall 5% higher Pearson Coefficient and 10% higher R-squared than a standard model trained on the same number of simulations;
- Comparable overall performance to a standard model trained on 14% fewer simulations;
- For QoI values above 83.5, the MSE is reduced by 60% (from Baseline 301) or 38% (from Baseline 349).

Final consolidated models and optimisation

Output consolidation, both using blade surface static pressure and exit flow properties, successfully improves the performance of the models. However, for all the compared models, the best accuracy is achieved by choosing as inputs only the geometric variables which have a stronger correlation with the QoI.

To facilitate the optimisation study, the consolidated predictions for the QoI are then used as supplementary inputs to models with a full set of geometric variables. The models "Optimise parametric" and "Optimise latent" have structure:

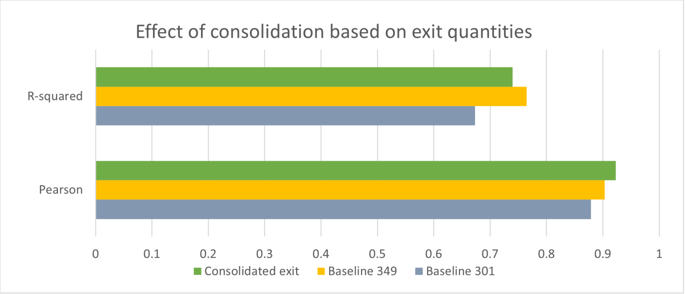

FIGURE 9. OVERALL PERFORMANCE OF OUTPUT CONSOLIDATION (FROM EXIT CONDITIONS).

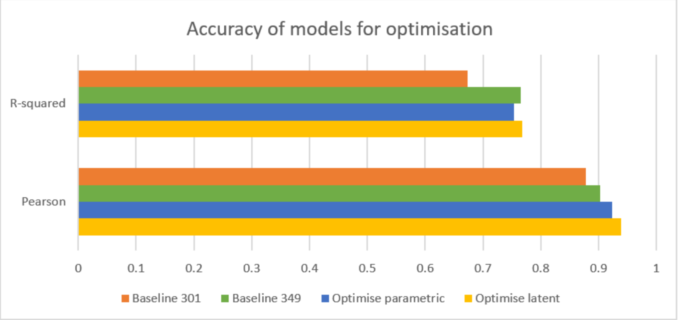

FIGURE 10. COMPARED ACCURACY OF BASELINE AND CONSOLIDATED MODELS WITH COMPLETE SET OF GEOMETRIC PARAMETERS AS INPUTS.

- Model: GPR with kernels *Matern1/2* and *Noise*;
- Inputs: predicted QoI from the two secondary models, 35 EDPs *or* 30 latent parameters;
- Output: QoI.

The performance of the two tertiary models is compared to that of the baseline models. The final improvement can be seen in Figure 10.

Figure 11 shows the predicted QoI against its actual value, as obtained from the two final models and from the baseline ones. The plot focuses on high values of the QoI, where the improvement in accuracy is particularly desirable in this optimisation setting.

Targeted optimisations of the two final consolidated models returned values for the optimal geometric parameters, EDPs or latent parameters.

The results of the confirmatory CFD simulation run on each optimised design are given in Table 1. The improvements in efficiency compare quite well with previous optimisation studies of NASA Rotor 37 found in the literature [29], especially for the optimisation based on the EDPs. The lower real-life performance of the blade described by the latent parameters might be partly due to the fact that the surface mesh generated by the decoder needs

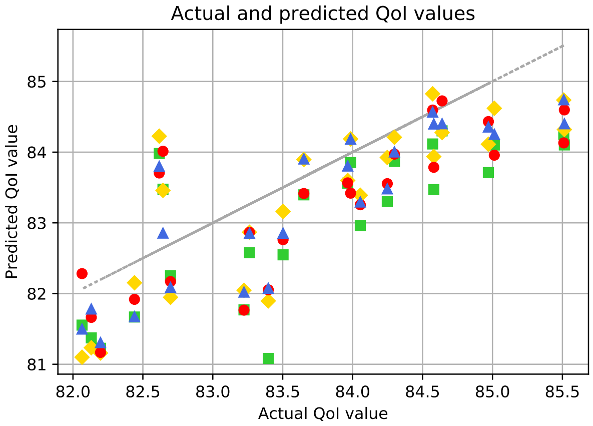

FIGURE 11. VALUES OF THE QOI AS PREDICTED BY BASELINE AND FINAL MODELS.

to first be converted back into EDPs and rebuilt using PADRAM to create a suitable mesh for the CFD run.

Note that the optimisation of the final models did not enforce any direct constraints on the pressure ratio (PR); the optimised geometries still fall within the threshold set for the quantity and are as such acceptable solutions to our optimisation study. However, it would be possible to develop a machine learning model to predict the PR from the available CFD data, in a similar way to what is presented here for the main QoI. The final model predicting the QoI would then take as an extra input the predicted PR and a constrained optimisation would return the optimal geometry which satisfies the pressure requirement.

CONCLUSIONS

This study investigated how the accuracy of machine learning models for turbomachinery applications can be increased by integrating predictions of relevant flow field quantities into the models for scalar performance indicators. For a specified training dataset, the combined use of autoencoders and output consolidation was able to achieve increased accuracy in the predictions, or alternatively it showed a performance comparable to that of models trained on 14% more simulations. The benefit offered by the proposed method therefore increases with the cost or complexity of the original simulations.

Of particular interest is the increased accuracy of the models for high values of the QoI, which are concerned in optimisation studies. For efficiency values above 83.5%, the MSE of the consolidated models was as much as 50% lower than seen in baseline models trained on 14% more simulations. As is common in machine learning, the model uncertainty grows when extrapolating to higher efficiencies than available in the training set (Table 1).

Optimisation of the QoI was carried out using the final models, which recovered the description of an optimal blade geometry in terms either of the EDPs or latent parameters from the autoencoder. Confirmatory CFD simulations were carried out on the optimised geometries, which showed significant improvements in the QoI of the blade optimised using the EDPs. The improvements achieved here compare favourably with values achieved by previous optimisation studies on NASA Rotor 37 available in the literature.

As the blades used for the CFD simulations are finally defined in terms of EDPs, the use of latent parameters in the optimisation stage was seen to partially offset the improved accuracy of the model, causing the CFD run to result in a lower efficiency than predicted by the model. However, the autoencoder still plays a key role in the presented analysis, as it allows much more accurate predictions of the intermediate quantities. As such, output consolidation would not be possible without the use of autoencoders.

Overall the method detailed here has the potential to accelerate the design of turbomachinery components, by reducing the number of (expensive and time-consuming) CFD simulations required to carry out meaningful optimisation studies. It also introduces the opportunity to reuse existing simulation data and extract from it information that is normally lost through direct optimisation of scalars.

ACKNOWLEDGMENT

The authors are grateful to Rolls-Royce plc for permission to publish this work. The authors would like to thank Diego Lopez for proving the results of CFD simulations used to train and test the models in this paper, and Pranay Seshadri for his comments about the work done. This study was made possible by a joint partnership between the University of Cambridge, MonolithAI and Rolls-Royce plc.

NOMENCLATURE

$\dot{m}$	Mass flow rate
η	Adiabatic efficiency, generally referred to as QoI.
P_0	Local exit total pressure
P_{exit}	Mass-average total pressure at the exit of the simulation domain
P_{inlet}	Total pressure at the inlet of the simulation domain
T_0	Local exit total temperature
T_{exit}	Mass-average total temperature at the exit of the simulation domain
T_{inlet}	Total temperature at the inlet of the simulation domain
DOE	Design of Experiments
EDP	Engineering Design Parameter, one of the 35 variables used to define the surface mesh geometry
GPR	Gaussian Process Regression

TABLE 1. RESULTS OF CFD SIMULATIONS FOR THE OPTIMISED GEOMETRIES.

Blade geometry	Predicted η (%)	η (%)	$\Delta\eta$	PR	Δ PR	training CFDs
Original geometry	–	84.406	–	2.081	–	–
Best in training set	–	86.649	+2.657%	2.082	+0.08%	–
Optimised EDPs	87.764	86.799	+2.835%	2.062	-0.92%	301 + 1
Optimised latent parameters	86.996	85.173	+0.909%	2.076	-0.3%	301 + 1

LE	Leading-edge of the blade
MAE	Mean Absolute Error
MSE	Mean Squared Error
NN	Neural Network
PADRAM	Parametric Design and Rapid Meshing
PR	Pressure Ratio across the blade
QoI	Quantity of Interest, the adiabatic efficiency of the blade, which is to be optimised
RANS	Reynolds-averaged Navier-Stokes equations
TE	Trailing-edge of the blade
URANS	Unsteady Reynolds-averaged Navier-Stokes equations

REFERENCES

[1] Brunton, S. L., Noack, B. R., and Koumoutsakos, P., 2020. “Machine learning for fluid mechanics”. *Annual Review of Fluid Mechanics,* ***52***, January, pp. 477–508.

[2] Azzam, M., and J. Haag, P. J., 2017. “Application concept of artificial neural networks for turbomachinery design”. *Computer Assisted Methods in Engineering and Science,* ***16***(2), January, pp. 143–160.

[3] Mondal, S., Joly, M. M., and Sarkar, S., 2019. “Multi-fidelity global-local optimization of a transonic compressor rotor”. In Proceedings of the ASME Turbo Expo 2019: Turbomachinery Technical Conference and Exposition, Vol. 2D: Turbomachinery, ASME. GT2019-91778.

[4] Umetani, N., and Bickel, B., 2018. “Learning three-dimensional flow for interactive aerodynamic design”. *ACM Transactions on Graphics,* ***37***(4), July, p. 89.

[5] Raveh, D. E., 2001. “Reduced-order models for nonlinear unsteady aerodynamics”. *AIAA Journal,* ***38***(8), August, p. 1417–1429.

[6] Shahpar, S., 2004. Design of experiment, screening and response surface modelling to minimise the design cycle time. VKI lecture series on Optimisation Methods Tools for Multicriteria/Multidisciplinary Design. LS-2004-08.

[7] Keane, A. J., 2006. “Statistical improvement criteria for use in multiobjective design optimization”. *AIAA Journal,* ***44***(4), April, p. 879–891.

[8] Kipouros, T., Molinari, M., Dawes, W., Parks, G., Savill, M., and Jenkins, K., 2007. “An investigation of the potential for enhancing the computational turbomachinery design cycle using surrogate models and high performance parallelisation”. In Proceedings of the ASME Turbo Expo 2007: Power for Land, Sea, and Air, Vol. Volume 6: Turbo Expo 2007, Parts A and B, ASME. GT2007-28106.

[9] Simpson, T., Mauery, T., Korte, J., and Mistree, F., 2001. “Kriging models for global approximation in simulation-based multidisciplinary design optimization”. *AIAA Journal,* ***39***(12), December, pp. 2233–2241.

[10] Jarrett, J., and Ghisu, T., 2012. “An approach to multi-fidelity optimization of aeroengine compression systems”. In 12thAIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14thAIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA. AIAA 2012-5634.

[11] Kontogiannis, S. G., Demange, J., Savill, A. M., and Kipouros, T., 2020. “A comparison study of two multifidelity methods for aerodynamic optimization”. *Aerospace Science and Technology,* ***97***.

[12] Akolekar, H. D., Zhao, Y., Sandberg, R. D., and Pacciani, R., 2020. “Integration of machine learning and computational fluid dynamics to develop turbulence models for improved turbine wake mixing prediction”. In Proceedings of ASME Turbo Expo 2020 Turbomachinery Technical Conference and Exposition GT2020, ASME. GT2020-14732.

[13] Hammond, J., Montomoli, F., Pietropaoli, M., Sandberg, R. D., and Michelassi, V., 2020. “Machine learning for the development of data driven turbulence closures in coolant systems”. In Proceedings of ASME Turbo Expo 2020 Turbomachinery Technical Conference and Exposition GT2020, ASME. GT2020-15928.

[14] Lav, C., and Sandberg, R. D., 2020. “Unsteady simulations of a trailing-edge slot using machine-learnt turbulence stress and heat-flux closures”. In Proceedings of ASME Turbo Expo 2020 Turbomachinery Technical Conference and Exposition GT2020, ASME. GT2020-14398.

[15] McCartney, M., Indlekofer, T., and Polifke, W., 2020. “Online detection of combustion instabilities using supervised

machine learning". In Proceedings of ASME Turbo Expo 2020 Turbomachinery Technical Conference and Exposition GT2020, ASME. GT2020-14834.

[16] Reid, L., and Moore, R. D., 1978. Performance of single stage axial-flow transonic compressor with rotor and stator aspect ratios of 1.19 and 1.26, respectively, and with design pressure ratio of 1.82. Available on the NASA Technical Reports Server.

[17] Hah, C., 2009. Large eddy simulation of transonic flow field in nasa rotor 37. In 47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition, p. 1061.

[18] Dunham, J., 1998. CFD validation for propulsion system components (la validation CFD des organes des propulseurs). Technical report, DTIC ADA349027.

[19] Chima, R., 2009. Swift code assessment for two similar transonic compressors. In 47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition, p. 1058.

[20] Polynkin, A., Toropov, V., and Shahpar, S., 2010. "Multidisciplinary optimization of turbomachinery based on metamodel built by genetic programming". In 13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA. Paper 2010-9397.

[21] Denton, J., 1997. "Lessons from rotor 37". *Journal of Thermal Science,* ***6***, November, pp. 1–13.

[22] Cumpsty, N., 2009. "Some lessons learned". In Proceedings of the ASME Turbo Expo 2009: Power for Land, Sea, and Air, Vol. 7: Turbomachinery, Parts A and B, ASME. GT2009-60368.

[23] Shahpar, S., Seshadri, P., and Parks, G., 2014. "Leakage uncertainties in compressors: The case of rotor 37". *Journal of Propulsion and Power,* ***31***, September.

[24] Shahpar, S., and Lapworth, L., 2003. "PADRAM: Parametric design and rapid meshing system for turbomachinery optimisation". In Proceedings of the ASME Turbo Expo 2003, collocated with the 2003 International Joint Power Generation Conference, Vol. 6: Turbo Expo 2003, Parts A and B, ASME. GT2003-38698.

[25] Milli, A., and Shahpar, S., 2012. "PADRAM: Parametric design and rapid meshing system for complex turbomachinery configurations". In Proceedings of the ASME Turbo Expo 2012: Turbine Technical Conference and Exposition, Vol. Volume 8: Turbomachinery, Parts A, B, and C, ASME. GT2012-69030.

[26] Tensorflow documentation. URL `https://www.tensorflow.org/guide/keras/sequential_model`.

[27] Sklearn documentation. URL `https://scikit-learn.org/stable/modules/gaussian_process.html`.

[28] Rasmussen, C. E., and Williams, C. K. I., 2006. *Gaussian Processes for Machine Learning*. The MIT Press, ISBN 026218253X.

[29] John, A., Shahpar, S., and Qin, N., 2016. "Alleviation of shock-wave effects on a highly loaded axial compressor through novel blade shaping". In Proceedings of the ASME Turbo Expo 2016: Turbomachinery Technical Conference and Exposition, Vol. Volume 2A: Turbomachinery, ASME. GT2016-57550.