Adapting turbofan noise modelling tool using neural networks

Proceedings of Global Power and Propulsion Society
ISSN-Nr: 2504-4400
GPPS Shanghai25
September 4-6, 2025
www.gpps.global

GPPS-TC-2025-014

Adapting turbofan noise modelling tool using neural networks

Chenzhao Li
chenzhao@chalmers.se
Chalmers University of Technology
Gothenburg, Sweden

Evangelia Maria Thoma
evangelia.maria.thoma@chalmers.se
Chalmers University of Technology
Gothenburg, Sweden

Xin Zhao
xin.zhao@chalmers.se
Chalmers University of Technology,
Gothenburg, Sweden

ABSTRACT

System-level multidisciplinary design and optimization (MDO) of turbofans, incorporating integrated noise and emissions predictions, can require substantial computational resources. Noise assessments typically consume significantly more compared to 1-dimensional simulations of turbofan performance and correlation-based emissions predictions, making them the bottleneck of the entire process. In this study, the use of neural networks in adapting an existing semi-empirical noise modelling tool for MDO applications is presented. The aim is to reduce the computation load while keeping the accuracy of the predictions sufficiently close to the original models. Deep neural networks (DNN) and its combination with K-nearest neighbors (KNN) make up stacking model have been applied for the purpose. The dataset for neural network training comes from the noise model developed by Chalmers Noise Code (CHOICE), an open-source framework with the capability to predict the source noise level, from individual airframe, engine components and the entire aircraft. For the aircraft approach phase, the neural network selects the important design parameters at the engine design point as inputs and outputs the noise of each engine component. Overall, the practical use of neural networks proves beneficial which could achieve noise prediction quickly and efficiently with high accuracy. The combined use of DNN and KNN could improve the accuracy of the trained models significantly.

INTRODUCTION

Since the introduction of turbofan engines in the 1950s, the commercial aviation industry has witnessed sustained and rapid growth. Alongside the rapid expansion of the aviation industry, noise pollution around airports has increasingly become a significant issue. The engine noise is considered to be one of the primary sources of jet aircraft noise, significantly affecting the airport staff and the people living or working in the vicinity (Zaporozhets and Tokarev, 1998; Mahashabde *et al.*, 2011; Zaporozhets, Tokarev and Attenborough, 2011; Delfs *et al.*, 2018). Modern prediction methods used in research for engine noise prediction are mainly based on semi-empirical or semi-analytical formulations, which are then transformed into far-field noise data through appropriate propagation models (Ventres, Theobald and Mark, 1982; Meyer and Envia, 1996; Dunn, Tweed and Farassat, 1999; Nark, Envia and Burley, 2009; Maria Thoma *et al.*, 2024). For turbofan engines, the major sources of noise components are compressor, fan inlet, fan discharge, combustor, turbine and jet (Matta, Sandusky and Doyle, 1977; Heidmann, 1979; Ribner, 1981; Mahan and Karchmer, 1991; Mendoza, Nance and Ahuja, 2008; Hultgren, 2010). During the early stages of an aircraft engine design, the performance parameters of these components under design conditions influence the magnitude of the above noise sources. Substantial computational resources and time are usually required for noise prediction based on semi-empirical or semi-analytical formulas due to the complexity of managing the entire engine design space under a multidisciplinary framework.

Machine learning methods, with their robust data processing capabilities, have been widely used in aeroacoustics prediction modelling (Revoredo, Mora-Camino and Slama, 2016; Tenney, Glauser and Lewalle, 2018; Akdeniz, Sogut and Turan, 2020; Li and Lee, 2020; Ikuta *et al.*, 2023; Redonnet *et al.*, 2024; Zhang *et al.*, 2024; Zhang, Yang and Zhang, 2024; Zhang and Zhang, 2024). They provide a new way to build an efficient and highly accurate engine noise prediction model. In this paper, machine learning methods are used to adapt an existing engine noise prediction model for a specific engine architecture across a wide range of design parameters. The aim is to create a fast-responding surrogated model to be used in system-level multidisciplinary design and optimization (MDO) of aircraft and engine design, providing reasonable noise predictions for various design options. To identify a suitable machine learning algorithm for adapting the existing engine noise prediction models, the performance of the Deep neural networks (DNN) algorithm, K-nearest neighbors (KNN) and the combined use of the two in predicting Effective Perceived Noise Level (EPNL) during the aircraft approach phase is compared.

DATASETS USED FOR THE MEACHING LEARNING MODEL DEVELOPMENT

The engine noise datasets used for machine learning model training were generated by the Chalmers Noise Code (CHOICE) (Maria Thoma *et al.*, 2024) which is an open-source framework with the capability of predicting the source noise level, for every frequency and longitudinal directivity, from individual airframe and engine components and the entire aircraft. The CHOICE noise estimation of aircraft engines was based on empirical and semi-empirical noise models found in published literature (Dunn and Peart, 1973; Fink, 1977; Fink, 1979; Heidmann, 1979; Russell, 1984; Kontos, Janardan and Gliebe, 1996; Sen *et al.*, 2004; Gliebe *et al.*, 2022). Every engine component was modelled separately, and the engine total noise was calculated as the sum of all the individual components.

The engine noise source estimation in CHOICE requires detailed engine performance parameters, which are estimated using the in-house code GESTPAN (Grönstedt, 2000). GESTPAN takes thrust, airspeed, altitude, and ambient conditions as input and generates several performance files for each of the engine components. The performance and dimension files for the engine were then input to CHOICE together with a trajectory file, and the engine noise was calculated for every point in the trajectory. The output from CHOICE was a sound pressure level matrix for every frequency and longitudinal directivity and for every trajectory point. This matrix was generated for every component separately, and the total noise was calculated by summing all the individual sources.

In this study, a case involving an A320-type aircraft was established. Turbofan design point parameters - Fan pressure ratio (FPR), Low-Pressure Compressor Pressure ratio (LPC PR), specific thrust, thrust, Mach number, deviation from International Standard Atmosphere (ΔISA), and altitude as listed in Table 1, have been varied around the nominal value within the specified range for this study. At the meantime overall pressure ratio, combustor outlet temperature and cooling flow fractions were fixed at design point. Specific thrust at design point was maintained for all cases except those with varying Specific Thrust. The thrust value at design point remained at the nominal value for all cases except those with varying thrust. A single key parameter was changed at a time, and changes in engine components and overall noise levels were stored and used as a dataset for subsequent machine learning training. The specified ranges for the selected parameters were defined to be able to demonstrate noticeable impact on noise levels while maintaining the parameters at reasonable levels. For example, a 2.5% change in design point FPR is relatively small compared to the changes of other parameters but can result in a 1 dB difference in engine total EPNL because of its significant impact on jet noise. The range for LPC PR variation, -15% to +15%, on the other hand, does not contribute to any clear difference in engine total noise but can activate the stage count change in LPC design and was initially considered as a good example to test KNN. After data generation, it has been found that most of the parameter variations have had stage count change in either LPC or LPT. The variation in design point ΔISA, as the only one which shows relatively linear relationship with the noise levels, was considered a good comparison case with the variations of other parameters in testing different machine learning models.

Table 1. Turbofan engine noise sensitivity parameters

Parameter	***Nominal Value***	***Range***
FPR	**1.37 [-]**	**-2.5% to +2.5%**
LPC PR	**2.8 [-]**	**-15% to +15%**
Specific Thrust	**244.5 [kg/s]**	**-7% to +7%**
Thrust	**21000 [N]**	**-20% to +20%**
Mach	**0.78 [-]**	**-10% to +10%**
Altitude	**10668 [m]**	**-10% to +10%**
ΔISA	**10.0 [K]**	**-10 K to +10 K**

*** Design point data if not specified**

MACHINE LEARNING MODELS

Deep Learning Model (DNN)

Deep Neural Network (DNN) is a machine learning model with multiple hidden layers between input and output. It processes data through interconnected neurons, applying activation functions and backpropagation to adjust weights, enabling complex pattern recognition and decision-making (LeCun, Bengio and Hinton, 2015; Goodfellow *et al.*, 2016). DNNs excel at handling complex data, recognizing patterns, and making accurate predictions. They improve with more data, support feature extraction, and perform well in regression tasks. Fig. 1 shows the configuration of the DNN model for predicting engine component noise and total noise, where the design point parameters as given in Table 1 were selected as inputs to train the DNN. A three-layer neural network model was used, with each hidden layer consisting of 64 nodes. The output includes six predicted engine component noise and total noise. The intermediate and output layers are fully connected. There are 705 design points used for training and testing.

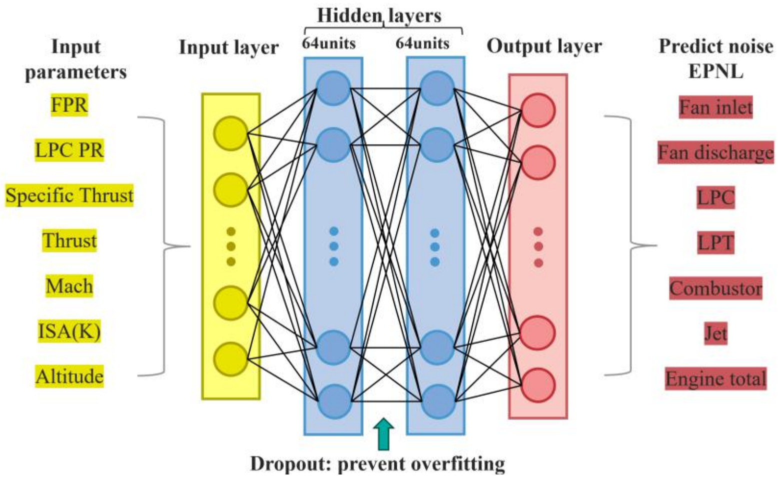


Fig.1 Schematic of Deep Learning Model

The rectified linear unit (ReLU) was used as the activation function. Dropout was applied to the output values of the first intermediate layer with a dropout rate of 0.3. Dropout is one of the most effective methods for preventing overfitting in neural networks (Baldi and Sadowski, 2013).The root-mean-square errors of the predicted engine noise EPNL for the six components and engine total (index *i*) with noise data amount (index *j*) are summarized to calculate the $RMSE_{total}$ shown as Eq. (1), where $E_{i,j}^{C}$ and $E_{i,j}^{P}$ are the CHOICE noise data and predicted noise data using NN.

$$RMSE_{total} = \sqrt{\frac{1}{560}\sum_{j=1}^{560}\frac{1}{7}\sum_{i=1}^{7}\left(E_{i,j}^{P} - E_{i,j}^{C}\right)^2} \qquad (1)$$

It is important to note that the raw database used here for training and testing the DNN is relatively small, compared to typical neural network applications. To minimize potential bias from the limited dataset, a 5-fold cross-validation method is employed. This technique ensures proper convergence during the training phase while using all data points to estimate network loss. As shown in Fig. 2, the entire dataset is first divided into two subsets according to the prescribed train-test split ratio. The first subset is used for DNN training, while the second is reserved for testing, following standard practice. However, the training subset is further divided into five non-overlapping sub-subsets. Four sub-subsets are used to train a specific DNN model, which is then evaluated using the fifth sub-subset. This process results in five independently trained and tested models, each following an 80%-20% train-test ratio. The best-performing fold is retained as the final DNN model and subsequently benchmarked against the unseen testing subset.

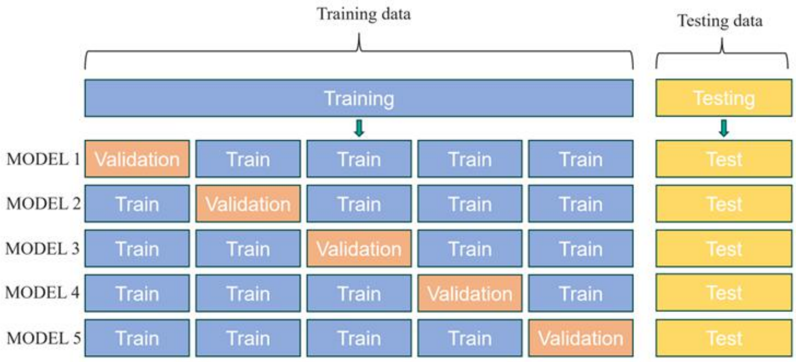


Fig.2 Sketch of the 5-fold cross-validation technique used in the DNN training.

The NN is trained to minimize the loss function $RMSE_{total}$ by using the Adam algorithm to update the weights (Kingma 2014). The backpropagation algorithm is used to calculate the gradient required by Adam. The number of epochs (number of times the weight coefficients are updated) is 2000. The model described above is implemented in Python with Keras library7 in Tensorflow backend (Abadi *et al.*, 2016).

Fig. 3 depicts the losses history of the DNN model, with illustration of the training phase and its convergence. As the training progresses, the loss function of the training dataset (blue line) decreases and becomes flat after 1500 epochs. The loss function for the testing dataset (red line) decreases and converges to approximately the same value as the blue line. There is no significant difference in the $RMSE_{total}$ between the training and test data, ranging from 0.1 to 0.2 dB. The results suggest that the DNN model achieve RMSE convergence in 2000 epochs.

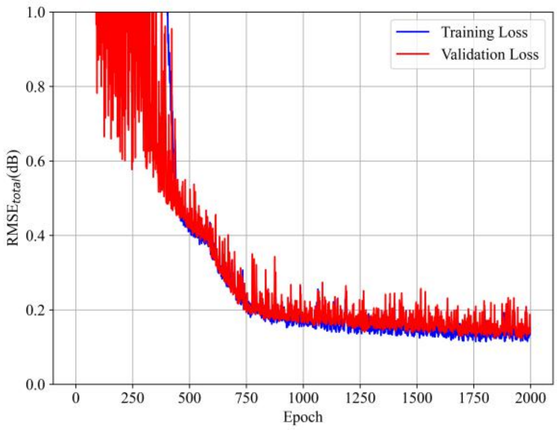


Fig.3 Learning curve of DNN

Stacking Algorithm (DNN-KNN)

KNN is a commonly used supervised learning algorithm (Guo *et al.*, 2003; Steinbach and Tan, 2009). Its primary aim is to calculate the distance between an object and its neighbors with known labels using Eq. 2:

$$d(x,y) = \left(\sum_{i=1}^{q} |\chi_i - y_i|^p \right)^{1/p} \tag{2}$$

Where x is the object, y is its neighbor with known label; when p is 1 and 2, the distance is Manhattan distance and Euclidean distance, respectively. Based on the computed distances, the algorithm identifies the k nearest neighbors, where k is a user-defined hyperparameter that significantly influences the performance of model. Predictions are then made by averaging the target values of these neighbors. For weighted KNN, a distance-based weighting scheme assigns greater influence on closer neighbors, thereby improving prediction accuracy in certain cases. One of the main advantages of KNN regression is its simplicity and ability to model non-linear relationships without requiring complex feature transformations. Additionally, it adapts well to datasets with local patterns, as predictions are derived from immediate neighbors information. In this case, we choose the number of neighbors is 3, the weighting method is distance and the parameter p is set to 2.

Ensemble learning can enhance notably the generalization performance of models. Its basic idea is to accomplish learning tasks by combining multiple learners. According to combination strategies, ensemble learning can be classified into stacking (Wolpert, 1992), bagging (Breiman, 1996) and boosting (Friedman, 2001). Bagging and boosting directly put several weak learners together to form a strong learner. The output of a strong learner is determined by averaging or voting on the outputs of weak learners. Unlike bagging and boosting, stacking utilizes a meta-learner to gather a group of base learners. The core concept of stacking lies in constructing a multi-layer learning framework, where the predictions from multiple base learners are used as input features and further combined by a meta-learner to enhance the model's generalization performance. The stacking method typically involves two primary stages: training the base learners and constructing the meta-learner. In the first stage, different base learners are independently trained on the same dataset and generate their respective predictions. These base learners can include models with varying bias-variance characteristics, such as decision trees, support vector machines, and neural networks. In the second stage, the predictions produced by the trained base learners are treated as new features and fed into the meta-learner. The meta-learner is tasked with combing the outputs of multiple base models to generate a final prediction to better fit the target variable. In this research, DNN and KNN are selected as base learners. The Support Vector Regression (SVR) model with a Gaussian kernel is a powerful nonlinear regression technique well-suited for capturing complex relationships between input features and target variables. By implicitly mapping the input data into a high-dimensional feature space, the model enables linear regression in that space to approximate nonlinear patterns in the original input space. This approach offers strong generalization capabilities, making it particularly effective for modeling high-dimensional, small-sample datasets. In this study, SVR is employed as the meta-model, with the penalty parameter C set to 2000, epsilon $\varepsilon = 0.01$, and the Gaussian kernel selected as the kernel function. Its process is demonstrated in Figure 4. Advantage of stacking lies in its ability to effectively reduce the bias and variance of individual models, thereby improving the overall robustness and generalization capability of the combined model.

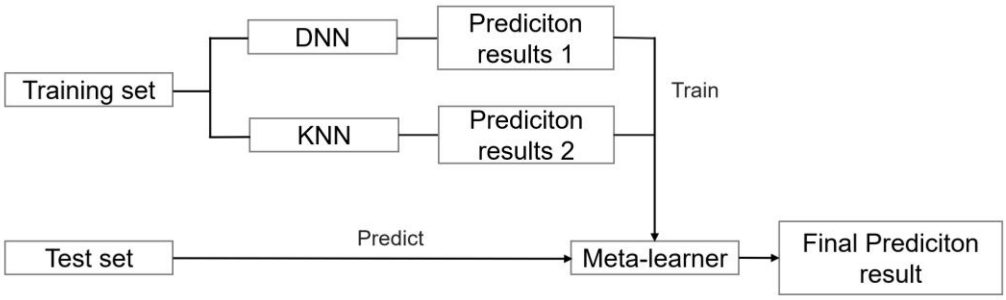


Fig.4 Diagram of the stacking model workflow

RESULTS AND DISCUSSION

Comparison of Machine Learning Algorithms in Noise Prediction

In this study, 20% of the total engine noise dataset was selected as a test set to evaluate the noise prediction performance of the DNN and stacking models. Fig. 5 illustrates the prediction performance of the three models on the test set for each engine component and the overall noise. The RMSE values of the three noise prediction models are less than 0.3 dB for each engine component and the overall noise. The highest RMSE value is approximately 0.29 dB for the LPC predicted by the DNN model. In comparison, the corresponding RMSE value for the stacking model is 0.17 dB, showing a significant reduction. Compared to the DNN and KNN models, the stacking model shows improvements in noise prediction for the LPC, LPT, and Jet components. In terms of noise prediction for the fan inlet, fan discharge, and combustor, the stacking model offers no substantial performance gain over KNN. Indeed, detailed comparison has shown that most of the benefits of the stacking model in the predictions are because of the combined use of KNN, while no substantial performance difference between the KNN and stacking models have been observed. Hence, also for clarity, the following analysis will focus on a comparison between the stacking and DNN models.

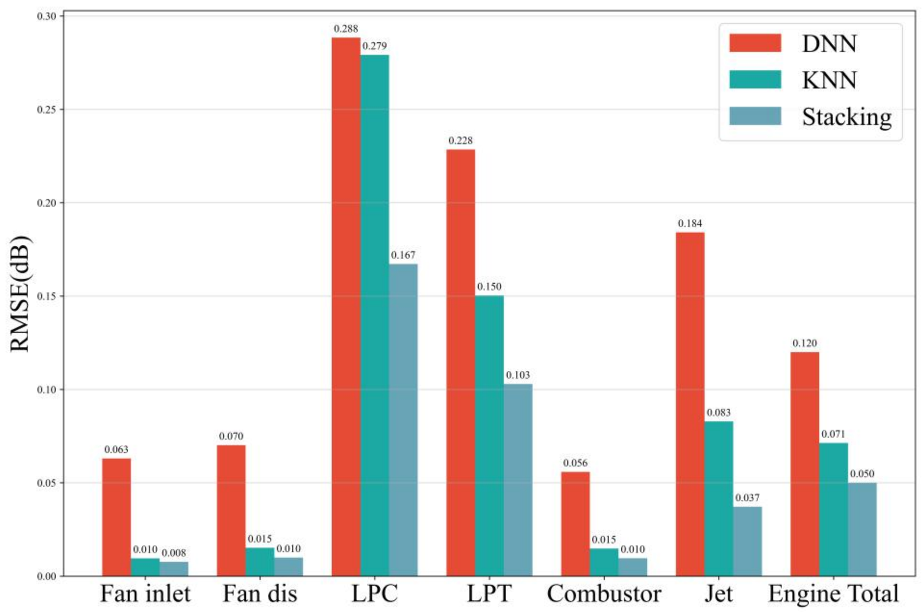
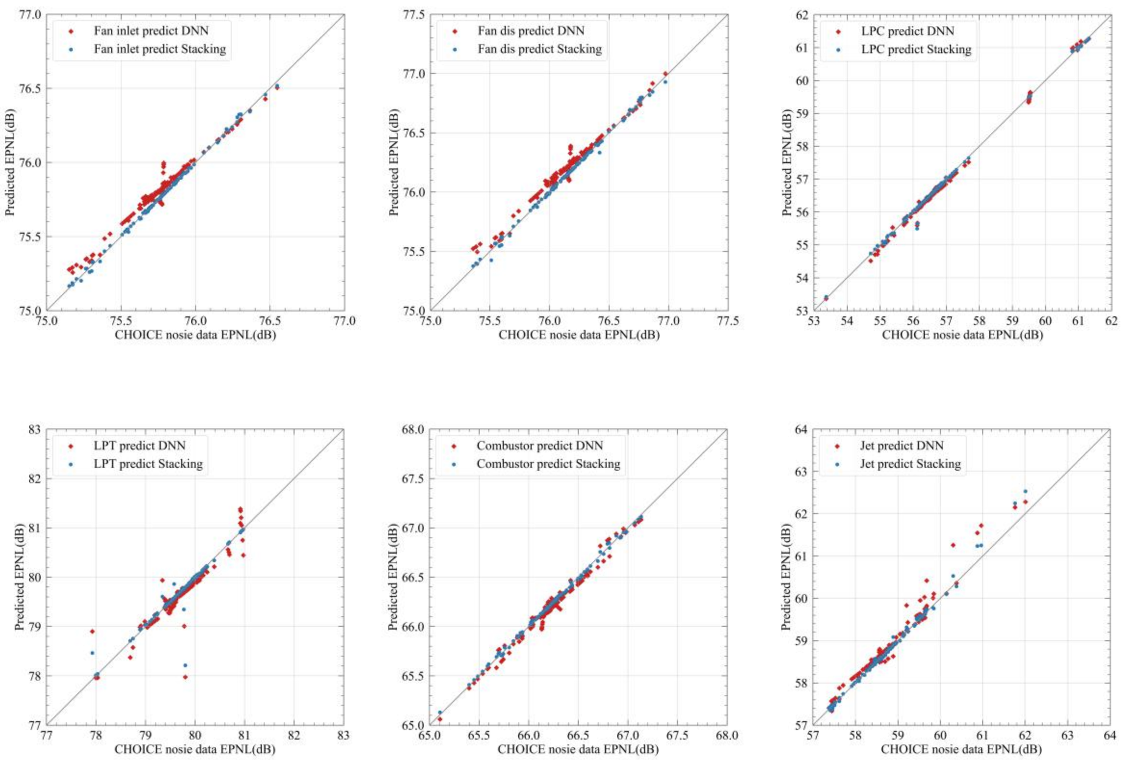


Fig. 5 The final loss function $RMSE_{total}$ values of testing dataset for each engine components and total.

The noise prediction results of stacking model and DNN model for each engine component are shown in Fig. 6. Both models demonstrate high accuracy in predicting the noise levels of the fan inlet, fan discharge, LPC, and combustor. The predicted values on the above engine components noise closely align with the CHOICE noise measurements, with error margins consistently within ±0.5 dB. Compared to the engine component noise mentioned above, the prediction errors of both models are slightly higher for the LPT, jet, and total engine noise. Both the DNN and stacking models exhibit some predictions with larger errors. However, overall, the stacking model demonstrates better convergence performance than the DNN model.



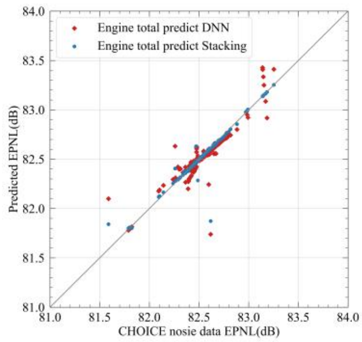


Fig.6 Comparison of all engine components EPNL prediction using DNN and Stacking Model

Engine Noise Predictions Sensitive Analysis

The sensitivities of key parameters on aircraft propulsion system noise assessment are investigated using an open-source aircraft noise prediction tool CHOICE as mentioned before. In addition, this section explores the performance of the two machine learning noise models compared to raw data when performing sensitivity analyses. Due to space limitations, altitude and Mach number were selected as the single variables for detailed analysis.

Figure 7 shows the noise variations of engine changes in altitude at the design point. Three different line styles were used to represent the CHOICE raw data, the DNN model predictions, and the stacking model predictions, respectively. The EPNL of fan inlet, fan discharge and combustor are almost unaffected by the variation of altitude. The LPC noise has an obvious step increase which is attributed to the increase in the number of compressor stages. Jet noise shows a downward trend due to the decrease in the mass flow rate decrease. As a result, Engine total noise remained as almost constant. Both the stacking model and the DNN model can predict the trend of EPNL variation. However, when faced with step changes, the stacking model predicts the trend more accurately than the DNN model, demonstrating higher overall accuracy.

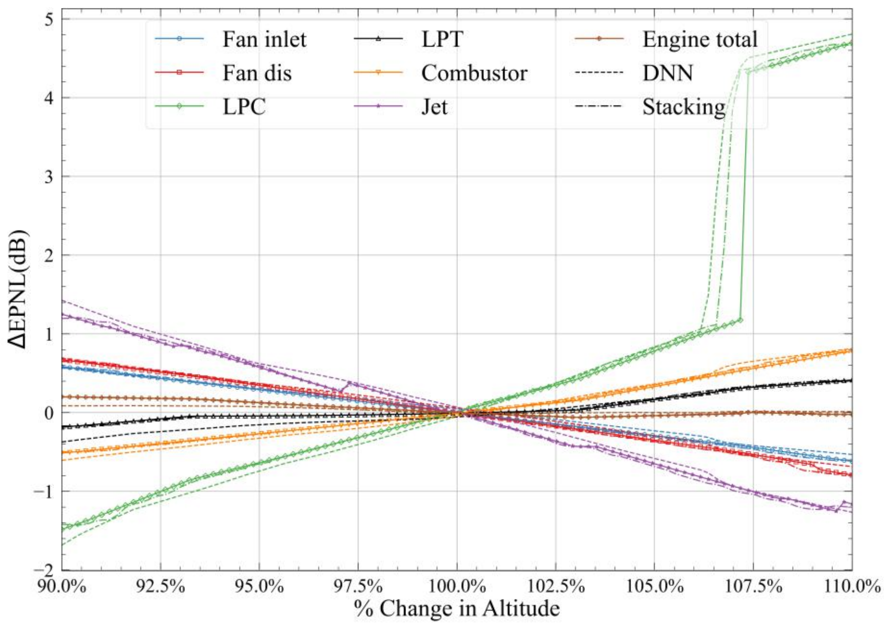


Fig.7 Variation of engine components and total noise EPNL for ±10% variation of altitude, relative to the baseline case

Figure 8 illustrates the noise variation for $\pm 10\%$ changes in flight Mach number. Fan and combustor noise showed a slight increase but remained almost constant. LPC noise showed a slight downward trend and LPT noise showed a clear step down but showed an increase after the step down occurred. Jet noise showed a clear drop followed by a gradual increase. Compared to the DNN model's more error-prone predictions, the stacking model's predictions align closely with the raw data, particularly in regions with step changes.

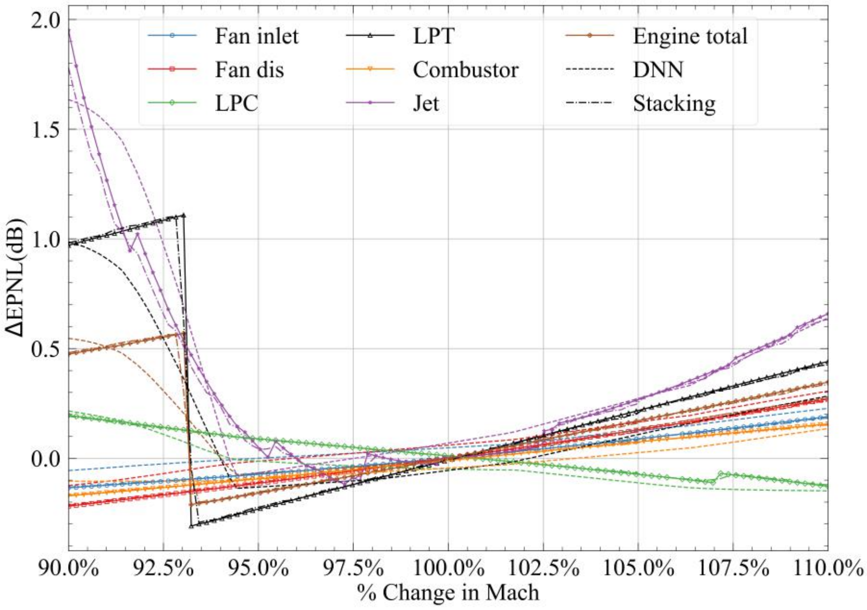


Fig.8 Variation of engine components and total noise EPNL for ±10% variation of Mach, relative to the baseline case.

The DNN model captures the global features and non-linear relationships of the data by fully utilizing the limited dataset through 5-fold cross-validation, demonstrating strong predictive capability. However, due to the limited amount of training data, the prediction accuracy of the DNN model decreases when step changes occur in the relationships between variables. In contrast, the KNN algorithm demonstrates better performance in these regions due to its inherent local approximation characteristics. KNN is a distance-based model, where predictions are based entirely on the distribution of neighboring points. The stacking model further enhances prediction accuracy by allowing the meta-function to perform a secondary integration of the DNN and KNN outputs, thereby achieving superior performance compared to either model alone.

CONCLUSIONS

In this paper, nonlinear aeroengine noise models with varying design parameters based on DNN and KNN algorithms are developed and applied to predict engine noise during the aircraft approach phase. The effect of variations in engine design point parameters on engine noise was investigated, and the performance of machine learning models was evaluated separately. In summary, the DNN and stacking models can accurately predict the noise generated by the engine during the approach phase. The RMSE values for each engine noise component on the test set are less than 0.3 dB. Compared to DNN model alone, the stacking model shows significant improvements in noise prediction for the LPC and LPT when handling step changes.

Compared to the original noise model which would take several minutes for noise prediction, the neural network adapted noise model has significantly reduced the computation time down to a few seconds while maintaining sufficient accuracy. The reduced prediction time offered by the deep neural network (DNN) is particularly valuable in the context of iterative design and optimization processes, where thousands of evaluations may be required to explore high-dimensional design spaces or perform real-time assessments. In such scenarios, cumulative time savings become substantial, making fast prediction a practical necessity rather than a convenience. However, it's worth noting that the training dataset used in this study is relatively small compared to typical machine learning applications. In future work, a more complete dataset will be utilized and the generated neural networks based noise model will be used in MDO framework for assessing its performance in engine and aircraft integrated design and optimization.

ACKNOWLEDGMENTS

This work has received funding from the Horizon Europe research and innovation programme under GA n° 101096275. This work is co-funded by UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee n° 10068673.

REFERENCE

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G. and Isard, M. (2016) *12th USENIX symposium on operating systems design and implementation (OSDI 16)*.

Akdeniz, H.Y., Sogut, M.Z. and Turan, O. (2020) 'Prediction and simulation of aircraft noise in the international Eskisehir Hasan Polatkan airport (LTBY)', *Aircraft Engineering and Aerospace Technology*, 93(1), pp. 171-179. Available at: 10.1108/aeat-03-2020-0056

Baldi, P. and Sadowski, P.J. (2013) 'Understanding dropout', *Advances in neural information processing systems*, 26.

Breiman, L. (1996) 'Bagging predictors', *Machine learning*, 24, pp. 123-140.

Delfs, J., Bertsch, L., Zellmann, C., Rossian, L., Kian Far, E., Ring, T. and Langer, S.C. (2018) 'Aircraft noise assessment—From single components to large scenarios', *Energies*, 11(2), p. 429.

Dunn, D.G. and Peart, N. (1973) *Aircraft noise source and contour estimation*.

Dunn, M., Tweed, J. and Farassat, F. (1999) 'The application of a boundary integral equation method to the prediction of ducted fan engine noise', *Journal of sound and vibration*, 227(5), pp. 1019-1048.

Fink, M.R. (1977) 'Airframe noise prediction method', *Final Report*.

Fink, M.R. (1979) 'Noise component method for airframe noise', *Journal of aircraft*, 16(10), pp. 659-665.

Friedman, J.H. (2001) 'Greedy function approximation: a gradient boosting machine', *Annals of statistics*, pp. 1189-1232.

Gliebe, P., Mani, R., Connell, S., Salamah, S., Sober, J. and Coffin, R. (2022) *AST Critical Propulsion and Noise Reduction Technologies for Future Commercial Subsonic Engines Aeroacoustic Prediction Codes-Supplement: Code Descriptions and Users Guides*.

Goodfellow, I., Bengio, Y., Courville, A. and Bengio, Y. (2016) *Deep learning (1)*. MIT press Cambridge.

Grönstedt, T. (2000) *Development of methods for analysis and optimization of complex jet engine systems*. Chalmers Tekniska Hogskola (Sweden).

Guo, G., Wang, H., Bell, D., Bi, Y. and Greer, K. (2003) *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*. Springer.

Heidmann, M. (1979) 'Interim prediction method for fan and compressor noise source', *NASA TX-71763 (June 1975)*.

Hultgren, L.S. (2010) *Core-Noise*.

Ikuta, Y., Imamura, T., Takaishi, T. and Kobayashi, T. (2023) 'Noise Prediction for Aircraft during Approach by Neural Network Using Sound Source Data Obtained from Microphone Array Measurement', *AIAA AVIATION 2023 Forum*. doi: 10.2514/6.2023-3927.

Kontos, K.B., Janardan, B. and Gliebe, P. (1996) *Improved NASA-ANOPP noise prediction computer code for advanced subsonic propulsion systems*.

LeCun, Y., Bengio, Y. and Hinton, G. (2015) 'Deep learning', *nature*, 521(7553), pp. 436-444.

Li, S. and Lee, S. (2020) 'A Machine Learning-Based Fast Prediction of Rotorcraft Broadband Noise', *Aiaa Aviation 2020 Forum*. doi: 10.2514/6.2020-2588.

Mahan, J.R. and Karchmer, A. (1991) 'Combustion and core noise', *Aeroacoustics of flight vehicles: theory and practice*, 1, pp. 483-517.

Mahashabde, A., Wolfe, P., Ashok, A., Dorbian, C., He, Q., Fan, A., Lukachko, S., Mozdzanowska, A., Wollersheim, C. and Barrett, S.R. (2011) 'Assessing the environmental impacts of aircraft noise and emissions', *Progress in Aerospace Sciences*, 47(1), pp. 15-52.

Maria Thoma, E., Grönstedt, T., Otero Sola, E. and Zhao, X. (2024) 'Assessment of an Open-Source Aircraft Noise Prediction Model Using Approach Phase Measurements', *Journal of Aircraft*, 61(3), pp. 745-760. Available at: 10.2514/1.C037332

Matta, R., Sandusky, G. and Doyle, V. (1977) *GE Core Engine Noise Investigation-Low Emmission Engines*. Department of Transportation, Federal Aviation Administration, Systems ….

Mendoza, J., Nance, D. and Ahuja, K. (2008) *14th AIAA/CEAS Aeroacoustics Conference (29th AIAA Aeroacoustics Conference)*.

Meyer, H.D. and Envia, E. (1996) *Aeroacoustic analysis of turbofan noise generation*.

Nark, D., Envia, E. and Burley, C. (2009) *15th AIAA/CEAS Aeroacoustics Conference (30th AIAA Aeroacoustics Conference)*.

Redonnet, S., Bose, T., Seth, A. and Li, L.K.B. (2024) 'Airfoil self-noise prediction using deep neural networks', *Engineering Analysis with Boundary Elements*, 159, pp. 180-191. Available at: 10.1016/j.enganabound.2023.11.024

Revoredo, T., Mora-Camino, F. and Slama, J. (2016) 'A two-step approach for the prediction of dynamic aircraft noise impact', *Aerospace Science and Technology*, 59, pp. 122-131. Available at: 10.1016/j.ast.2016.10.017

Ribner, H.S. (1981) 'Perspectives on jet noise', *AIAA journal*, 19(12), pp. 1513-1526.

Russell, J.W. (1984) *An empirical method for predicting the mixing noise levels of subsonic circular and coaxial jets*.NASA.

Sen, R., Hardy, B., Yamamoto, K., Guo, Y. and Miller, G. (2004) 'Airframe noise sub-component definition and model'.

Steinbach, M. and Tan, P.-N. (2009) 'kNN: k-nearest neighbors', in *The top ten algorithms in data mining*. Chapman and Hall/CRC, pp. 165-176.

Tenney, A.S., Glauser, M.N. and Lewalle, J. (2018) 'A Deep Learning Approach to Jet Noise Prediction', *2018 AIAA Aerospace Sciences Meeting*. doi: 10.2514/6.2018-1736.

Ventres, C., Theobald, M. and Mark, W. (1982) *Turbofan noise generation. volume 1: Analysis*.

Wolpert, D.H. (1992) 'Stacked generalization', *Neural networks*, 5(2), pp. 241-259.

Zaporozhets, O., Tokarev, V. and Attenborough, K. (2011) *Aircraft Noise: Assessment, prediction and control*. CRC Press.

Zaporozhets, O.I. and Tokarev, V.I. (1998) 'Aircraft noise modelling for environmental assessment around airports', *Applied Acoustics*, 55(2), pp. 99-127.

Zhang, Q., Wang, X., Yang, D. and Zhang, W. (2024) 'Data-driven prediction of aerodynamic noise of transonic buffeting over an airfoil', *Engineering Analysis with Boundary Elements*, 163, pp. 549-561. Available at: 10.1016/j.enganabound.2024.04.006

Zhang, Q., Yang, D. and Zhang, W. (2024) 'Optimal transonic buffet aerodynamic noise PSD predictions with Random Forest: Modeling methods and feature selection', *Aerospace Science and Technology*, 150 Available at: 10.1016/j.ast.2024.109245

Zhang, Q. and Zhang, W. (2024) 'A hybrid approach for reconstruction of transonic buffet aerodynamic noise: Integrating random forest and compressive sensing algorithm', *Aerospace Science and Technology*, 152 Available at: 10.1016/j.ast.2024.109379

APPENDIX A - COPYRIGHT/OPEN ACCESS

APPENDIX B - GPPS Presenter Policy and Paper Acceptance

According to GPPS's presenter attendance policy, a paper cannot be published or indexed and may not be cited as a published paper until at least one author pays the registration fee and attends the conference. The GPPS reserves the right to withdraw from its publications any paper that is not presented by an author of the paper at the appropriate conference. Any paper that is withdrawn may not be cited as a published paper.