A Pareto Front Based Evolutionary Model for Airfoil Self-Noise Prediction

Amirhessam Tahmassebi†, Amir H. Gandomi*, and Anke Meyer-Baese†
†Department of Scientific Computing, Florida State University, Tallahassee, Florida 32306-4120, USA
Email: atahmassebi@fsu.edu , URL: www.amirhessam.com
*School of Business, Stevens Institute of Technology, Hoboken, New Jersey 07030, USA
Email: a.h.gandomi@stevens.edu

***Abstract*—According to NASA's report on the technologies that could reduce external aircraft noise by 10 dB, a challenge equally as important as finding approaches on airframe noise reduction is the demand to bring up strategies by which airframe noise can be predicted both accurately and rapidly. One of the components of the overall airframe noise is the self-noise of the airfoil itself. In this paper, an evolutionary symbolic implementation for airfoil self-noise prediction was proposed. Multi-objective genetic programming as a subset of evolutionary computation along with adaptive regression by mixing algorithm was used to create an executable fused model. The developed model was tested on the airfoil self-noise database and the performance of the developed model was compared to the previous works and benchmark machine learning algorithms. The reasonable results suggest that the proposed model can be applied to noise generation by low-Mach-number turbulent flows in aerospace, automobile, underwater, and wind turbine acoustic communities.**

I. Introduction

Recently, numerous researches have been conducted on reducing aircraft engine noise at take off [1][2][3]. In 2004, NASA published a report to identify the technologies that could reduce external aircraft noise by 10 dB [4]. This requires reducing the acoustic power by 90%. In 1977, Fink [5] started research into airframe noise prediction and proposed an empirically based airframe noise prediction model. One of the component of the overall airframe noise is the self-noise of the airfoil itself. The airfoil that passes through smooth non-turbulent inflow conditions and interacts with boundary layers near wake region causes the self-noise [6]. There have been various researches conducted to predict airfoil self-noise including light-hill acoustic analogy, linearized hydrodynamic equations, and semi-empirical models [7]. Moreau et al. [8] conducted an analysis of flow conditions in free-jet experiments for studying airfoil self-noise. In addition to this, Geyer et al. [1] measured the noise generation at the trailing edge for various porous airfoils. A challenge equally as important as finding approaches on airframe noise reduction is the demand to bring up strategies by which airframe noise can be predicted precisely [4].

In 1989, Brooks et al. [9] implemented a semi-empirical model to predict self-noise of airfoil using an extensive set of acoustic wind tunnel tests employing various chord length of `NACA0012` airfoil sections. The noise generated by turbulent boundary layer mechanism at high Reynolds number flow conditions. In this condition, the boundary layer over the airfoil is turbulent, which causes noise once the turbulence goes through the trailing edge of the airfoil and encounters the free-stream flow. They collected a database of input variables based on the frequency of the noise, the relative angle of attack to wind, the free-stream velocity, and the geometric parameters of the airfoil including the displacement thickness and chord length. They have used a scaling rule to scale the sound pressure level as the output of the experiment [9][10]. Fig. 1 depicts the airfoil nomenclature diagram in details.

In this paper an evolutionary symbolic implementation for airfoil self-noise prediction was proposed. In this implementation various models with different goals using the combination of multi-objective genetic programming and adaptive regression by mixing algorithm were proposed. The proposed models were compared to the benchmark machine learning algorithms along with the model proposed by Brooks et al. [9] in 1989 which is still utilized for noise prediction in wind turbine designs. The proposed model can be applied to noise generation by low-Mach-number turbulent flows in aerospace, automobile, underwater, and wind turbine acoustic communities [8].

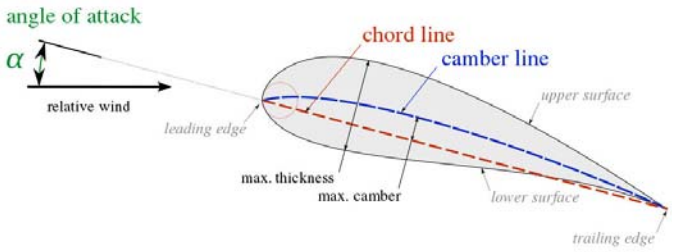


Fig. 1. Airfoil nomenclature diagram.[1]

II. Genetic Programming

Genetic Programming (GP) [11] as a symbolic optimization technique searches the space of mathematical expressions to find the model that best fits a given dataset, both in terms of accuracy and simplicity. GP due to the selection of designs applies on fitness and complexity measurements

[1]https://en.wikipedia.org/wiki/Airfoil

978-1-5090-6017-7/18/$31.00 ©2018 IEEE

Algorithm 1: NSGA-II

Input: Generations N, Population P
Output: Best Model

Initialize population P;
Generate random population size M;
Evaluate objective values;
Assign ranking based on Pareto sort;
Generate child population;
Binary tournament selection;
Recombination and mutation;
for $i \in \{1, \ldots, N\}$ **do**
  for *each Parent and Child in Population* **do**
    Assign ranking based on Pareto sort;
    Generate sets of non-dominated solutions;
    Determine Crowding distance;
    Loop inside by adding solutions to next generation starting from the first front until N individuals;
  end
  Select points on the lower front with higher crowding distance;
  Create next generation;
  Binary tournament selection;
  Recombination and mutation;
end

Algorithm 2: ARM

Input: Input Variables X_i, Target Variables Y_i, i ∈ (1,N), Function $\hat{f}$
Output: Best Model

Random permutation the order of the observations M;
for $m \in \{1, \ldots, M-1\}$ **do**
  Randomly permute the order of the observations.;
  Split the data into two parts ;
  $Z^{(1)} = (X_i, Y_i)_{i=1}^{\frac{N}{2}}$;
  $Z^{(2)} = (X_i, Y_i)_{i=\frac{N}{2}+1}^{N}$;
  for $j \in \{1, \ldots, J\}$ **do**
    Estimate $\hat{f}_{j,\frac{N}{2}}(x; Z^{(1)})$ of f ;
    Estimate the variance function $\sigma^2(x)$ by $\hat{\sigma}^2_{j,\frac{N}{2}}(x)$;
    for $i \in \{\frac{N}{2}+1, \ldots, N\}$ **do**
      Predict Y_i by $\hat{f}_{j,\frac{N}{2}}(X_i)$;
    end
    $E_j = \frac{\prod_{i=\frac{N}{2}+1}^{N} h((Y_i - \hat{f}_{j,\frac{N}{2}}(X_i))/\hat{\sigma}_{j,\frac{N}{2}}(X_i))}{\prod_{i=\frac{N}{2}+1}^{N} \hat{\sigma}_{j,\frac{N}{2}}(X_i)}$;
    Compute the current weight $\hat{W}_j = \frac{E_j}{\sum_{l=1}^{J} E_l}$;
  end
  The final estimate is $\hat{f}_N(x) = \sum_{j=1}^{J} \hat{W}_j \hat{f}_{j,N}(x)$;
end

phase was formulated originally based on functional programming language as an evolutionary method to use computer programs for solving a problem following the principle of Darwinian natural selection. GP instead of using one candidate, uses a group of individuals (population), which are formed by stochastically combining mathematical building blocks such as mathematical operators, analytic functions, constants, and state variables and genetic operators to make new individuals (generations) guided by objective functions such as fitness and complexity which measure the quality of each individual. GP function regressor [12][13] as shown in Algorithm 1 was implemented as a Multi-Objective Genetic Programming (MOGP) approach based on Non-Dominated Sorting Genetic Algorithm II (NSGA-II) introduced by Deb at al. [14]. The algorithm minimizes both the error (mean-absolute-error (MAE) or mean-square-error (MSE)) of the models and the subtree complexity measure [15] as the non-linearity order of the model simultaneously [12][13][16]. GP has shown great performance in predicting complex patterns using its evolutionary nature [17][18][19][20] and flexibility to be combined with parallel algorithms to run multiple jobs using high performance computing (HPC) [21].

The proposed model forms a Pareto front of models based on the fitness and subtree complexity measures. In this study, the least complex model, the most accurate model, and the model at the knee of the Pareto front were presented. Additionally, a fused model of the Pareto front obtained with Adaptive Regression by Mixing (ARM) algorithm introduced by Yang [22] was presented. Algorithm 2 illustrates the details of the ARM implementation.

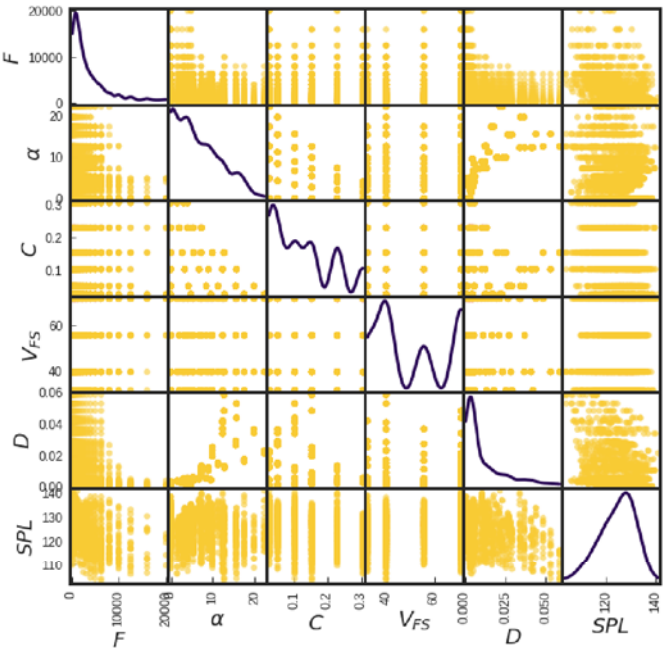
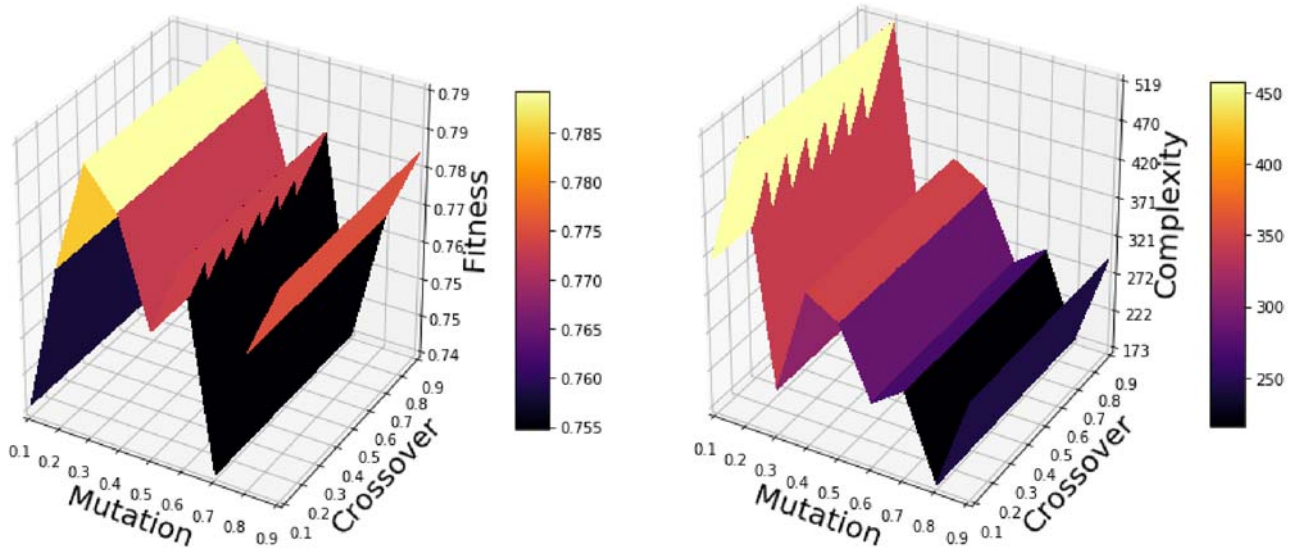


Fig. 2. Scatter matrix representation of the input/output variables in the airfoil self-noise database with kernel density function estimations of the input/output variables on the diagonal.



(a) Fitness (b) Complexity

Fig. 3. Exploring the optimized values for mutation and crossover rates for the developed GP model based on the fitness and complexity measures.

III. Airfoil Self-Noise Database

Brooks et al. [9] collected a noise spectrum database from a family of NACA0012 airfoils based on turbulent boundary layer noise mechanism. The data base consists of 5 input variables including: (1) frequency (F) in Hertz (Hz), (2) angle of attack (α) in degree $(^\circ)$, (3) chord length (C) in meters (m), (4) free-stream velocity (V_{FS}) in meters per second (m/s), and (5) suction side displacement thickness (D) in meters (m). The scaled sound pressure level (SPL) in decibel (dB) was used as the output variable, which is implemented as follows:

$$Scaled \quad SPL_{1/3} = SPL_{1/3} - 10\log(\frac{M^5 DL}{r_e^2}) \qquad (1)$$

where M is the Mach number, L is the airfoil span, D is the suction side displacement thickness, and r_e is the retarded observer position. Fig. 2 depicts the scatter matrix presentation of the input/output variables with kernel density function estimations. The database contains 1503 exemplars, which was divided into training and testing sets based on the chord length values. The training set contains 5 different values for chord length including $[0.0254, 0.0508, 0.1524, 0.2286, 0.3048]$ in meters. The exemplars with the chord length of 0.1016 (m) were used as the testing set. Thus, the training set and the testing set comprise 1240 and 263 exemplars, respectively [6][9][23].

IV. Results & Discussion

A GP model based on multi-objective genetic programming approach and NSGA-II was developed. Total five features in the database were used as the inputs of the GP model for the symbolic regression task. To optimize the mutation and the crossover rates of the model, various rates ranging from 0.1 to 0.9 were employed for the developed GP model for 200 generations. Fig. 3 illustrates the 3-dimensional surface plot of the mutation rates and the crossover rates based on the fitness and subtree complexity measures. The highest fitness value was obtained with a value of 0.7 as the crossover rate, and a value of 0.3 as the mutation rate. Additionally, the lowest complexity was obtained with a value of 0.3 as the crossover rate, and a value of 0.7 as the mutation rate. However, the complexity that was obtained with values of 0.3 and 0.7 as the mutation rate and the crossover rate, respectively were considered as the second lowest. Therefore, these values

TABLE I
Parameters setting for the GP function regressor.

Parameter	Setting
Population Size	1000
Number of Generations	5000
Tournament Size	20
Number of Inputs	5
Crossover Rate	0.7
Mutation Rate	0.3
Number of Exemplars in Training Set	1240
Number of Exemplars in Testing Set	263
Number of CPU Threads	8
1^{st} Objective	MSE
2^{nd} Objective	Subtree Complexity
Population Initialization	Ramped-Half-and-Half
Function Set	$+, -, \times, /, \sqrt{\ }, (\ )^2, (\ )^3, (\ )^4$ $\log, \exp, \sin, \cos$

were considered to run the developed GP model with 5000 generations and 1000 population for the regression task. The details of the parameters setting for GP function regressor is presented in Table I. The developed GP model was trained on the training dataset with 1240 exemplars and tested on the testing dataset with 263 exemplars. This would decrease the chance of over-fitting of the model.

At each generation, the fitness and the complexity values were reported for the best individuals. Fig. 4 depicts the evolution of the GP model through generations for both fitness and complexity measures. The left Y-axis was set to fitness measure, the right Y-axis to complexity, and X-axis to number of generations. As seen, after roughly about 100 generations the model reached the fitness with 80% accuracy. After 5000 generations the fitness reached a value of 88% with the complexity value of 7560 which is about eight times more than the value that was obtained for the 80% fitness measure. By increasing the number of generations both fitness and complexity values increased as well.

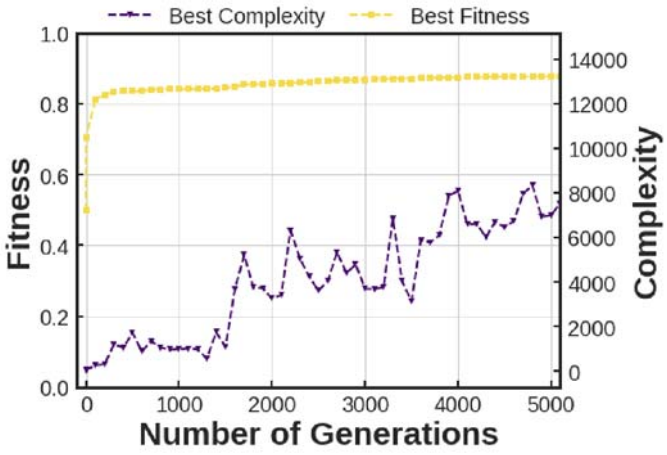


Fig. 4. The evolution of the employed objective functions, fitness and complexity measures for the developed GP model through different numbers of generations.

By considering the combinations of the fitness and complexity measures three different models were defined. The first model, which is the least complex model was obtained at the first generation with the subtree complexity of 5 and fitness value of 0.4963. The second model, which is the most accurate model was obtained after 4850 generations with a fitness value of 0.878 and complexity measure of 6911. The third model is the knee model, which was the point at the model that the slope of the fitness and complexity measures did not change through evolution of the generations. The knee model was found after 92 generations with a fitness value of 0.8124 and subtree complexity value of 197. Thus, the knee model can be a great alternative to be used in the regression task with 98% less computational run-time. In addition to this, a fused model based on ARM algorithm as shown in Algorithm 2 was presented as well. The proposed fused model based on the ARM algorithm has the ability of adapt itself over

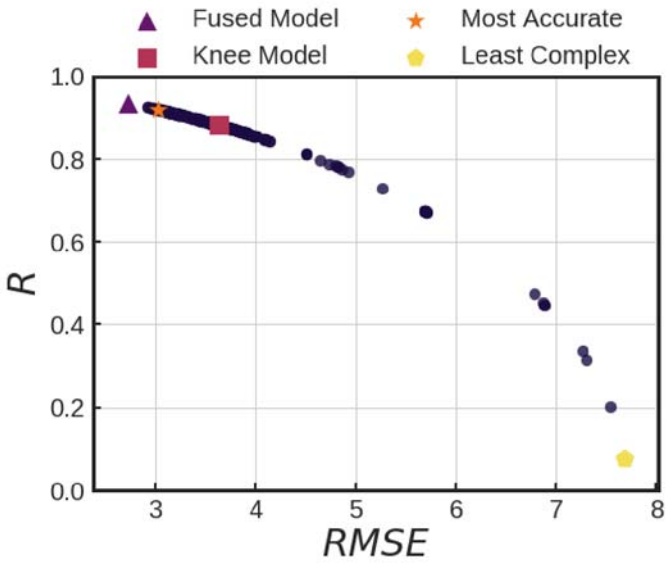
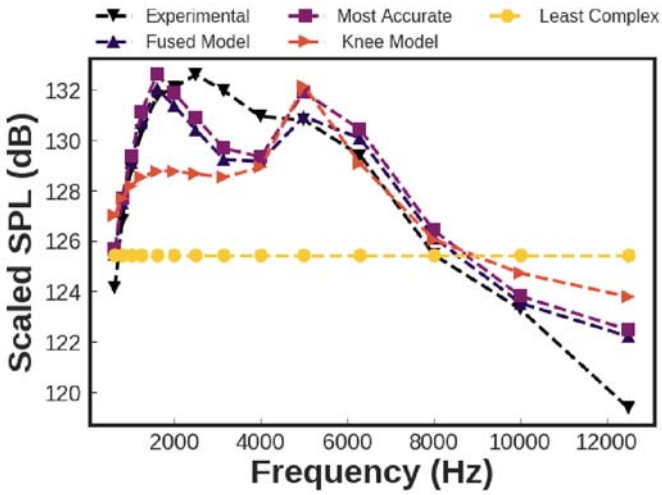


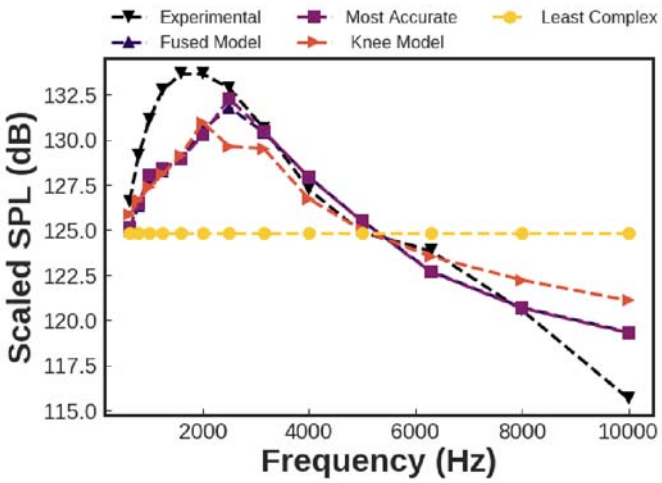
Fig. 5. Scatter plots of coefficient of determination versus root-mean-square error (R vs $RMSE$) of the all models in the Pareto front (dark violet circles) with indicating the most accurate model (orange star), the least complex model (yellow pentagon), the fused model (purple triangle), and the knee model (maroon square).

different procedures to perform well under various conditions. In other words, the goal of employing the ARM algorithm was to produce a model by giving different weights to some of the models in the Pareto front via proper assessment of performance of the estimators [22].

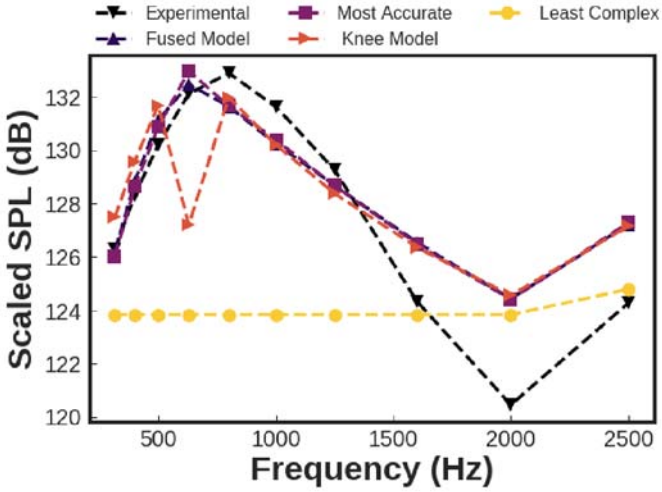
In the context of statistical modeling with the main purpose of prediction of future outcomes, coefficient of determination (R^2) provides a measure of how well observed outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model. Additionally, root-mean-square error ($RMSE$) is frequently used to measure the differences between values predicted by a model and the values actually observed [24]. To explore the performance of the proposed models, both regression metrics were computed for all of the models in the Pareto front (shown in dark violet circles) as shown in Fig. 5. Additionally, the specified models including the least complex model (shown in yellow pentagon), the most accurate model (shown in orange star), the knee model (shown in maroon square), and the fused model (shown in purple triangle) were indicated. It is clearly shown that the ARM algorithm found a model even better than the most accurate model in the Pareto front. The fused model with a value of 0.9452 as R, a value of 7.44 as MSE, and a value of 2.09 as MAE outperformed the other proposed models in the Pareto front including the most accurate model with a value of 0.9194 for R, a value of 9.18 for MSE, and a value of 2.28 for MAE metrics. It should be noted that the knee model has also shown reasonable performance with a value of 0.8819 for R, a value of 13.18 for MSE, and a value of 2.72 for MAE. However, these number were obtained only after less than 200 generations. In fact, to save the compu-



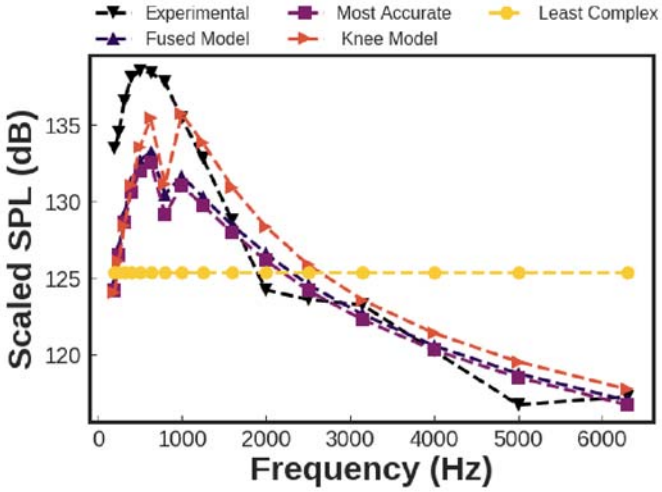
(a) $\alpha = 0°$



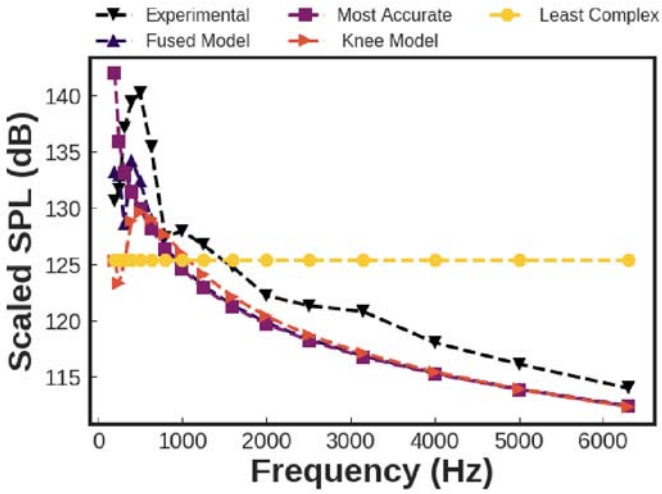
(b) $\alpha = 3.3°$



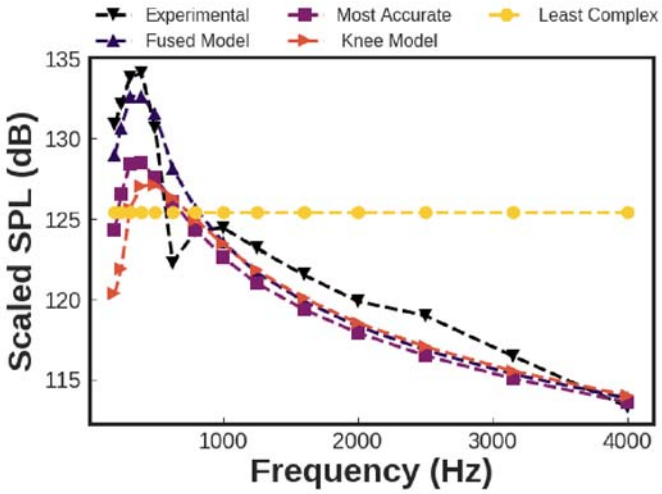
(c) $\alpha = 6.7°$



(d) $\alpha = 8.9°$

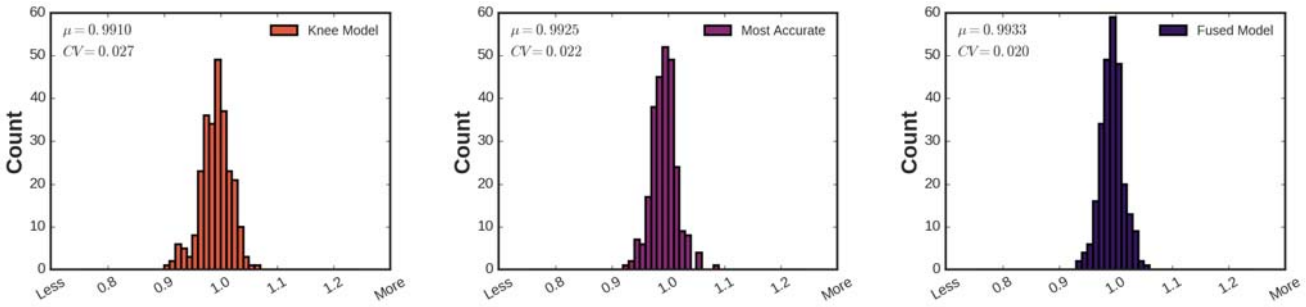


(e) $\alpha = 12.3°$



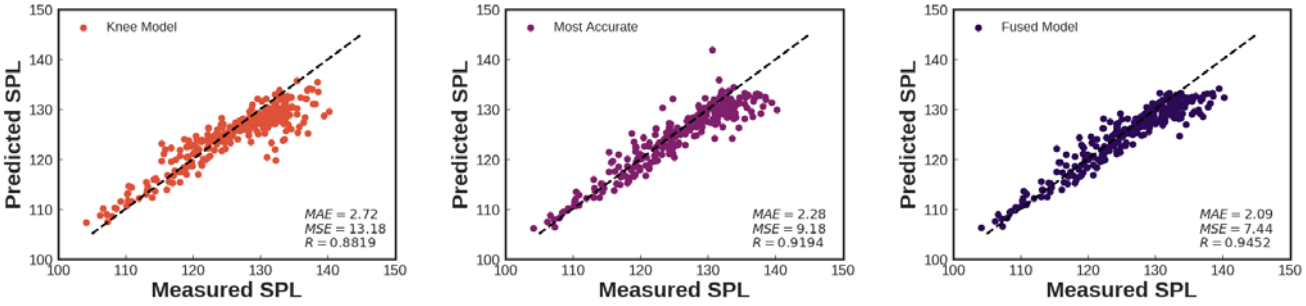
(f) $\alpha = 15.6°$

Fig. 6. An exhaustive comparison of the predicted scaled SPL at different frequencies for the most accurate model, the least complex model, the fused model, and the knee model versus the experimental data at different angles of attack and fixed chord length at 10.16 cm.



(a) Knee Model (b) Most Accurate Model (c) Fused Model

Fig. 7. Histograms of the ratio of the predicted and the measured SPL for (a) the most accurate model, (b) the knee model, and (c) the fused model. Mean value and coefficient of variation of this ratio are also reported for each model.



(a) Knee Model (b) Most Accurate Model (c) Fused Model

Fig. 8. Measured versus predicted SPL values using (a) the most accurate model, (b) the knee model, and (c) the fused model. The black dashed line indicates the ideal fit.

tational run-time, the knee model would be a great choice as an alternative to the most accurate model. Additionally, the performance of the selected models from the Pareto front were compared with the experimental data as shown in Fig. 6. In this regard, an exhaustive comparison of the predicted scaled SPL at different frequencies for different angles of attack including $\alpha = [0°, 3.3°, 6.7°, 8.9°, 12.3°, 15.6°]$ and a fixed chord length at $C = 10.16\ cm$ was done. It is clearly shown that all the spikes occurred for the frequencies less than $2000\ Hz$. That would be considered as the challenging part of the regression since the developed models could not predict the spikes of the experimental data for the small angles of attack except for the $\alpha = 15.6°$ as shown in Fig. 6f using the fused model. The illustrations of the predicted values versus the experimental values put lights on the facts that the models should be revised to capture the spikes. Even the most accurate model had still linear nature and could not predict very promising the extrema point in the data where the slope changed. However, employing the ARM algorithm and giving weights to some of the models in the Pareto front as the fused model showed better performance as shown in Fig. 6f. The results of the proposed models outperformed the results presented by Brooks et al. [9] and Lau et al. [6]. The model presented by Brooks et al. under-fitted the SPL values at both low and high frequencies. Fig. 7 presents the histograms of the ratio of the predicted and the measured SPL for different different models. In order to study the quality assurance, mean (μ) and coefficient of variation (CV) of this ratio are also reported. As shown, the mean values of all the histograms are approximately equal to one. The histogram of the fused model presents the best mean value of 0.9933. Employing coefficient of variation of a model helps to have better understanding of standard deviation of data in the context of the mean value of the data. Additionally, Fig. 8 demonstrates the measured versus predicted SPL values using the developed models. As an alternative to the developed GP model based on NSGA II, Multi-Stage Genetic Programing (MSGP) can be employed to reduce the error decomposition [25][26]. The MSGP algorithm consists of two main stages: (1) incorporating the individual effect of the input variables, (2) incorporating the interactions among the input variables. The MSGP formulates these two terms in an efficient procedure to optimize the error among predicted and actual values. This procedure can be done with employing parallel processing algorithms to run multiple jobs at the same time with high performance computing.

As the final assessment, the performance of the most common machine learning regression models including: (1) Gradient Boosting (GB), (2) Random Forest (RF), (3) Decision Tree (DT), (4) Support Vector Regression (SVR), and (5) Least Absolute Shrinkage and Selection Operator (LASSO) were compared to the developed GP model [24]. All the models were implemented in Python [27] and were trained on the

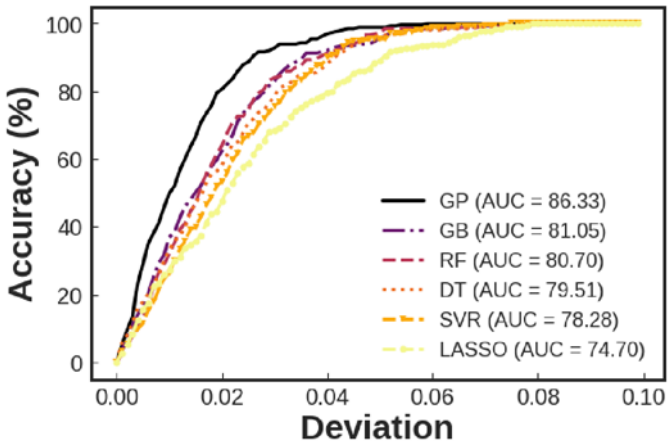


Fig. 9. REC curves of various regression methods along with calculated AUC compared to the developed symbolic regression model using GP.

training set, and were validated on the testing set. In machine learning, a Receiver Operating Characteristic (ROC) curve visualizes the performance of a classifier applied on a binary class problem across all possible trade-offs between the false positive rates and the true positive rates [24]. A graph consists of multiple ROC curves of different models characterizes the performance of the models on a binary problem and makes the comparison process of the models easier by visualization. Additionally, the area under the ROC curve (AUC) represents the expected performance of the classification model as a single scalar value [28].

Although ROC curves are limited to classification problems, Regression Error Characteristic (REC) curves can be used to visualize the performance of the regressor models. REC illustrates the absolute deviation tolerance versus the fraction of the exemplars predicted correctly within the tolerance interval. The resulting curve estimates the cumulative distribution function of the error. The area over the REC curve (AOC), which can be calculated via the area under the REC curve (AOC = 1 - AUC) is a biased estimate of the expected error. Furthermore, the coefficient of determination R^2 can be also calculated with respect to the AOC [28]. Likewise the ROC curve, the shape of the REC curve can also be used as a guidance for the users to reveal additional information about the data modeling. The REC curve was implemented in Python[2] and the details of the error metrics and scaling of the residuals are also available [29]. Fig. 9 demonstrates the REC curves of the implemented machine learning algorithms in comparison to the GP model. As shown, all of the models predicted all the exemplars with a normalized deviation of 0.08 correctly. By decreasing the deviation tolerance, the fraction of the correct predictions decreased as well. A value of 0.05 can be the critical tolerance for the employed models. The developed GP model with an AUC of 86.33% outperformed the other models. In fact, the developed GP model has the ability of prediction of the exemplars with a deviation of less than 0.03 precisely. In addition to this, Fig. 10 illustrates the radar plot of the performance of the regression models and the model proposed by Brooks et al. [9] in terms of coefficient of determination R. By comparing the coefficient of determination, it is clear that the GP model with an R value of 0.9452 outperformed the other models. The closest R score to the proposed model is GB with an R score of 0.8446. It should be noted that the model proposed by Brooks et al. with an R score of 0.8087 is still utilized for noise prediction in wind turbine engines. However, the proposed model showed a reasonable potential as an alternative to the proposed model by Brooks et al.

[2]https://github.com/amirhessam88/Regression-Error-Characteristic-Curve

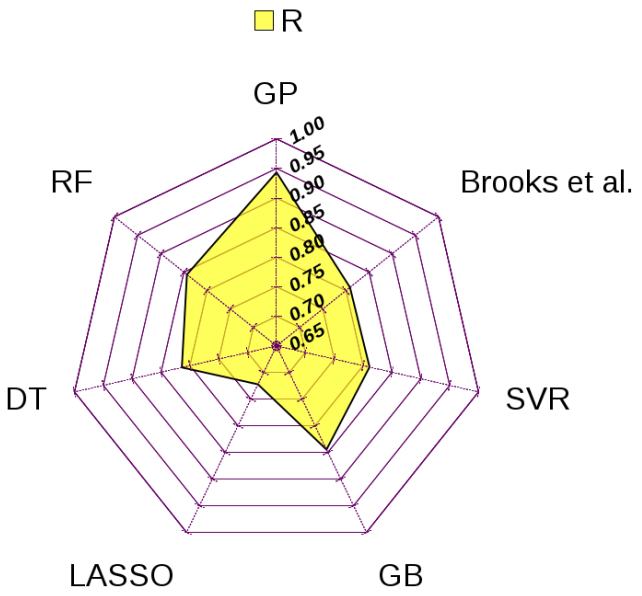


Fig. 10. Radar plot representation of the coefficient of determination for various regression methods compared to the developed symbolic regression model using GP.

V. Conclusions

This paper aims at developing an evolutionary symbolic implementation for airfoil self-noise prediction. In this regard, a multi-objective genetic programming strategy based on non-dominated sorting genetic algorithm II with considering the optimization of mean-square error as the fitness measure and the subtree complexity as the complexity measure simultaneously was employed. The GP model ran for 5000 generations with 1000 population considering training/testing sets to overcome any possible over-fitting. Additionally, the mutation and crossover rates were optimized over 200 generations based on the fitness and complexity measures. A total of five features from `NACA0012` noise database were employed as the inputs of the GP function regressor. Table II presents the summary statistics including correlation coefficient (R),relative root-mean-square error ($RRMSE$), relative mean absolute error ($RMAE$), and performance index (PI) of the results of the GP function regressors including the most accurate model, the

least complex model, the knee model, and the fused model. Higher R values and lower $RRMSE$ values result in lower PI, and consequently, indicate a more precise model. It should be noted that PI values range from 0 to $+\infty$ and the proposed performance index in Table II indicates the model predicts the actual values very well.

TABLE II
REGRESSION SCORE METRICS OF THE SELECTED MODELS IN THE PARETO FRONT.

Model	R	$RRMSE$	$RMAE$	PI
Knee	0.8819	2.89%	2.16%	0.015
Fused	0.9452	2.17%	1.66%	0.011
Most Accurate	0.9194	2.41%	1.82%	0.012
Least Complex	0.0758	6.10%	5.17%	0.056

The developed GP model was compared with the other machine learning algorithms. It was clearly shown that the GP model outperformed the other machine learning algorithms applied on this database. Moreover, the developed model was compared with the model proposed by Brooks et al. in 1989 which is still utilized for noise prediction in wind turbine designs. The reasonable performance of the developed model suggests that the proposed evolutionary approach can be applied to similar works including noise generation by low-Mach-number turbulent flows in aerospace, automobile, underwater, and wind turbine acoustic communities.

ACKNOWLEDGMENTS

The authors would like to thank Eitan Lees for the careful revision of the manuscript.

REFERENCES

[1] T. Geyer, E. Sarradj, and C. Fritzsche, "Measurement of the noise generation at the trailing edge of porous airfoils," *Experiments in Fluids*, vol. 48, no. 2, pp. 291–308, 2010.

[2] C. R. Ilario da Silva, T. H. Orra, and J. J. Alonso, "Multi-objective aircraft design optimization for low external noise and fuel burn," in *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2017, p. 1755.

[3] L. Bertsch, B. Schäffer, and S. Guerin, "Towards an uncertainty analysis for parametric aircraft system noise prediction," 2017.

[4] D. P. Lockhard and G. M. Lilley, "The airframe noise reduction challenge," 2004.

[5] M. R. Fink, "Airframe noise prediction method," United Technologies Research Center East Hartford CT, Tech. Rep., 1977.

[6] K. Lau, R. López, E. Oñate, E. Ortega, R. Flores, M. Mier-Torrecilla, S. Idelsohn, C. Sacco, and E. González, "A neural networks approach for aerofoil noise prediction," *Master thesis*, 2006.

[7] M. S. Howe, "A review of the theory of trailing edge noise," *Journal of sound and vibration*, vol. 61, no. 3, pp. 437–465, 1978.

[8] S. Moreau, M. Henner, G. Iaccarino, M. Wang, and M. Roger, "Analysis of flow conditions in freejet experiments for studying airfoil self-noise," *AIAA journal*, vol. 41, no. 10, pp. 1895–1905, 2003.

[9] T. F. Brooks, D. S. Pope, and M. A. Marcolini, "Airfoil self-noise and prediction," 1989.

[10] R. Schlinker, "Airfoil trailing edge noise measurements with a directional microphone," in *4th Aeroacoustics Conference*, 1977, p. 1269.

[11] J. R. Koza, *Genetic programming: on the programming of computers by means of natural selection*. MIT press, 1992, vol. 1.

[12] K. Veeramachaneni, I. Arnaldo, O. Derby, and U.-M. OReilly, "Flexgp," *Journal of Grid Computing*, vol. 13, no. 3, pp. 391–407, 2015.

[13] K. Veeramachaneni, O. Derby, D. Sherry, and U.-M. O'Reilly, "Learning regression ensembles with genetic programming at scale," in *Proceedings of the 15th annual conference on Genetic and evolutionary computation*. ACM, 2013, pp. 1117–1124.

[14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[15] E. Y. Vladislavleva *et al.*, *Model-based problem solving through symbolic regression via pareto genetic programming*. CentER, Tilburg University, 2008.

[16] J. McDermott, D. R. White, S. Luke, L. Manzoni, M. Castelli, L. Vanneschi, W. Jaskowski, K. Krawiec, R. Harper, K. De Jong *et al.*, "Genetic programming needs better benchmarks," in *Proceedings of the 14th annual conference on Genetic and evolutionary computation*. ACM, 2012, pp. 791–798.

[17] A. H. Gandomi, A. H. Alavi, and C. Ryan, *Handbook of genetic programming applications*. Springer, 2015.

[18] A. Tahmassebi, A. H. Gandomi, I. McCann, M. H. Schulte, L. Schmaal, A. E. Goudriaan, and A. Meyer-Baese, "An evolutionary approach for fmri big data classification," in *Evolutionary Computation (CEC), 2017 IEEE Congress on*. IEEE, 2017, pp. 1029–1036. [Online]. Available: http://doi.org/10.1109/CEC.2017.7969421

[19] A. Tahmassebi and A. H. Gandomi, "Building energy consumption forecast using multi-objective genetic programming," *Measurement*, 2018. [Online]. Available: https://doi.org/10.1016/j.measurement.2018.01.032

[20] A. Tahmassebi, A. H. Gandomi, I. McCann, M. H. Schulte, L. Schmaal, A. E. Goudriaan, and A. Meyer-Baese, "fmri smoking cessation classification using genetic programming," in *Workshop on Data Science meets Optimization*, 2017. [Online]. Available: http://ds-o.org/images/Workshop_papers/Gandomi.pdf

[21] A. Tahmassebi, A. H. Gandomi, and A. Meyer-Bäse, "High performance gp-based approach for fmri big data classification," in *Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact*, ser. PEARC17. New York, NY, USA: ACM, 2017, pp. 57:1–57:4. [Online]. Available: http://doi.acm.org/10.1145/3093338.3104145

[22] Y. Yang, "Adaptive regression by mixing," *Journal of the American Statistical Association*, vol. 96, no. 454, pp. 574–588, 2001.

[23] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[24] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. Wiley, New York, 1973.

[25] A. H. Gandomi and A. H. Alavi, "Multi-stage genetic programming: a new strategy to nonlinear system modeling," *Information Sciences*, vol. 181, no. 23, pp. 5227–5239, 2011.

[26] A. Tahmassebi and A. H. Gandomi, "Genetic programming based on error decomposition: A big data approach," in *Genetic Programming Theory and Practice XV*. Springer, 2018.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[28] J. Bi and K. P. Bennett, "Regression error characteristic curves," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003, pp. 43–50.

[29] A. Tahmassebi, "ideeple: Deep learning in a flash," in *Disruptive Technologies in Information Sciences*, vol. 10652. International Society for Optics and Photonics, 2018.