Advances in Computational Design, *Vol. 5, No. 3 (2020) 291-304*
DOI: https://doi.org/10.12989/acd.2020.5.3.291

Predicting the 2-dimensional airfoil by using machine learning methods

K. Thinakaran*[1], R. Rajasekar[2a], K. Santhi[3b] and M. Nalini[4c]

[1]*Computer Science Engeneering., Saveetha School of Engineering, SIMATS, Chennai 600 077 TN, India*
[2]*Aeronautical Engineering, MVJ Engineering College, Bangalore, India*
[3]*Sreenivasa Institute of Technology and Management Studies, Chittoor, India*
[4]*Computer Science Engeneering., Saveetha School of Engineering, SIMATS, Chennai 600 077 TN, India*

(Received November 5, 2018, Revised May 24, 2019, Accepted February 4, 2020)

Abstract. In this paper, we develop models to design the airfoil using Multilayer Feed-forward Artificial Neural Network (MFANN) and Support Vector Regression model (SVR). The aerodynamic coefficients corresponding to series of airfoil are stored in a database along with the airfoil coordinates. A neural network is created with aerodynamic coefficient as input to produce the airfoil coordinates as output. The performance of the models have been evaluated. The results show that the SVR model yields the lowest prediction error.

Keywords: support vector regression model; neural networks; airfoil design; inverse design; backpropagation

1. Introduction

The aim of this paper is to generate geometry for airfoil with minimal error. Hence, we are supposed to enhance design method which will converge fast and have minimal error. There are many techniques in use, for example, hodo-graph methods for two-dimensional flows, (Garabedian 1971). The inverse design of backpropagation takes long time to converge. Hertz (1991) focused on better function and suitable learning rate and momentum in backpropagation algorithm. To design fast algorithm, Abid *et al.* (2001) proposed a new algorithm by minimizing sum of squares of linear and nonlinear errors for all output. Jeong *et al.* (2001) proposed learning algorithm based on first and second order derivatives of neural activation at hidden layers. Han *et al.* proposed modified constrained learning algorithms—First new Learning Algorithm and Second new Learning Algorithm to obtain faster convergence rate. The notable differences of the accelerated modified LM method are that the line search for the approximate LM step, Jinyan Fan (2014).

One of important things is when designing a particular neural network is to calculate proper weight for neural activities. The weight values are obtained from the training process of neural

*Corresponding author, Ph.D., E-mail: thinakarank.sse@saveetha.com
[a] Ph.D., E-mail: rajasekar080564@yahoo.com
[b] Ph.D., E-mail: santhiglorybai@gmail.com
[c] Ph.D., E-mail: nalini.tptwin@gmail.com

http://www.techno-press.org/?journal=acd&subpage=7 ISSN: 2383-8477 (Print), 2466-0523 (Online)

network. To obtain appropriate weight in a neural network design utilizes two set of equations. First, the neural network equation is used to calculate the error function. The feedback neural network equation is next used to calculate the gradient vector. The use of gradient vector is for defining search directions in order to calculate weight change. The traditional inverse method uses the conformal mapping of flow domains.

The present inverse design is an alternative design procedure based on artificial neural networks. The CFD solver is first used to generate solutions which are stored and serve as an input to the neural network. The neural network is trained based on stored data. The neural network is then ready to perform airfoil design procedures limited by the training data. This is the technique for design of Airfoil based on non-parametric mapping using Neural Networks. A neural network uses supervised learning to train the neural network structure. Selviah (1991) showed the improvements gained by Generalised Correlation Higher Order Neural Networks.

Two methods are typically used in studies: analytical and simulation modelling. Analytical models become intractable, and are not practical for application dependent studies. Simulation is a feasible approach that can produce an accurate picture of Airfoil. Statistical method can characterize the behaviour of the program with some probability distributions. The important benefit is that a synthetic trace is very small network. The neural network is trained based on stored data. The neural network is then ready to perform airfoil design procedures limited by the training data. This is the technique for design of Airfoil based on non-parametric mapping using Neural Networks. A neural network uses supervised learning to train the neural network structure. Selviah (1991) showed the improvements gained by Generalised Correlation Higher Order Neural Networks. Machine learning techniques earned much importance for the prediction of the various parameters in different fields of science and engineering (Susom Dutta 2018).

Two methods are typically used studies: analytical and simulation modelling. Analytical models become intractable, and are not practical for application dependent studies. Simulation is a feasible approach that can produce an accurate picture of Airfoil. Statistical method can characterizes the behavior of the program with some probability distributions. The important benefit is that a synthetic trace is very small compared to real program traces (Genbrugge 2007). A statistical simulation is a robust tool in Airfoil design. Statistical simulation is still be time consuming especially when the systems to be simulated have many parameters and these parameters have to be tested with different probability distributions. Some studies in literature (Zayid 2012), which prove the fact that artificial intelligence method could be applied to perform simulation. The dataset contained the following input variables: An airfoil profile can be described by a set of 'x' and 'y' coordinates.

Support Vector Regression (SVR) was used to build prediction models. It was concluded that SVR model is a promising tool for predicting the Airfoil structure. Additional studies with different programming models are definitely required in order to generalize the effectiveness of machine learning methods. In this paper SVR and MFANN have been employed to predict the airfoil. The results show that the SVR model has the lowest error value. The aeroelastic behavior is sensitive in an Airfoil design process. Publications can be found in the area of uncertainty quantification for aero elasticity problems (Pettit 2004). The boundaries are created using a support vector machine (SVM). SVM belongs to the class of classifiers and is widely used in the computer science community (Cristianini 2002). The SVM boundary is constructed from an initial design of experiment whose samples are classified into categories based on the response of the system.

The set of results concerns the construction of explicit LCO boundary for the airfoil problem.

The LCO boundary is subsequently included in an design optimization problem with a constraint. Among the nonlinear phenomena, limit cycle oscillations (LCOs) have emerged as an interesting design challenge Kousen (1994) and Dowell (2002). Other studies on aeroelastic design optimization can also be found by Allen (2004) and Maute (2003). The rest of this paper is organized as follows: Section 2 summarizes the inverse design of airfoil. Section 3 presents an overview of MFANN and SVR. Section 4 is devoted to the details of models. Section 5 presents results and discussion. Finally, Section 6 concludes the paper.

2. Inverse design of airfoil

The inverse methods use the conformal mapping of the flow domains. It can be used for two-dimensional (2D) potential and Euler flows in which the flow equations can be transformed. On the other hand, the optimization methods aim at minimizing some objective function characteristic of the airfoil performance. The flow equation is taken as one kind of aerodynamic constraints in the optimization procedure. These methods are not only used for 2D, potential and Euler flows, but also for Navier Stokes flows. This method has become more and more popular in the last decade. For the inverse problem, the desired pressure distributions on the airfoil surfaces are specified. The summation of squares of difference between the actual (pi) and target (pit) pressures is used as the objective function F (fitness) which is described as follows in equation (1).

$$F = \sum_{i=1}^{N} (p_i - p_{it})^2 \tag{1}$$

where N is the number of grid point on the airfoil surfaces. In the present inverse design procedure based on artificial neural networks is performed. The CFD solver is first used to generate solutions which are stored and serve as an input to the neural network. The backpropagation neural network is trained based on these data using a suitable algorithm and the optimized network is determined. The neural network can perform any design procedures subject to the input range limited by the training data. An aircraft design depends on the multidisciplinary factors such as aerodynamic efficiency, structural weight, stability and control. A design is finalized only after numerous iterations. The development of accurate methods for aerodynamic shape optimization is the goal of optimal design.

In our research, an airfoil is drawn with 26x and 26y coordinates with corresponding coefficient of lift (CL) and the coefficient of drag (CD). The database consists of 78 NACA series airfoils with different combinations of CL, CD, x and y coordinates. Here the CL, CD and x coordinates as input values to the neurons in the input layer of ANN. The y coordinates are considered as output value in the output layer of ANN. We are going to calculate the y coordinates using the proposed algorithm. The calculated y coordinates should match with the y coordinates stored in the database. This is the "inverse" problem for airfoil design.

3. Overview of methods

3.1 Multilayer feed-forward neural networks

The feed forward backpropagation network is a supervised training; it has finite number of pattern pairs consisting of an input pattern and a desired output pattern. In the training, an input pattern is presented at the input layer. Then neurons pass the pattern to the hidden layer. The outputs of the hidden layer neurons are obtained through activation function with bias value or threshold value. The Hidden layer outputs value become inputs to the output neurons, which process the inputs value using an optional bias and a threshold function. At last final output of the network is determined by the activation function from the output layer.

The computed pattern and the stored pattern are compared. The error is difference between computed pattern and stored pattern. Based on this, adjustment to weights of connections between the hidden layer and the output layer are computed. Similarly, based on the error in the output, change is made for the connection weights between the input and hidden layers. The procedure is repeated for the entire stored pattern. The pass through all the training patterns is called an epoch. The process is repeated as many cycles as needed until the error is within a prescribed tolerance limit. From the optimization literature that gradient descent methods are usually very inefficient. In gradient descent, if search space contains long ravines then it results in oscillation which makes it difficult to find search direction. Gill handles this situation by modifying the search direction with introduction of momentum term.

In training, the energy function has to be minimized. If the network achieves the optimum value in a finite number of steps, then you have for the operation of the network. In that case, to avoid a lot of computation time, introduce a momentum parameter to change the weight in order to speed up the convergence. Momentum is a portion of the previous weight change added to current weight. Momentum improves the rate of convergence. Change of weight is described in Fig. 1.

The momentum also overcomes the effect of local minima. Sejnowski and Rosenberg Parker (1987) proposed a similar momentum method that used exponential smoothening. In the 1990s Rumelhart put effort into popularizing the training algorithm among the neural network scientific community. Presently, the backpropagation algorithm is also used for training of other categories of neural networks. In the training, neural networks aims to find a correct set of weights that give us a global minimum in the error function. Fu (1994) illustrated that the error surface of neural networks is generally described to be complex, convex and contains concave regions.

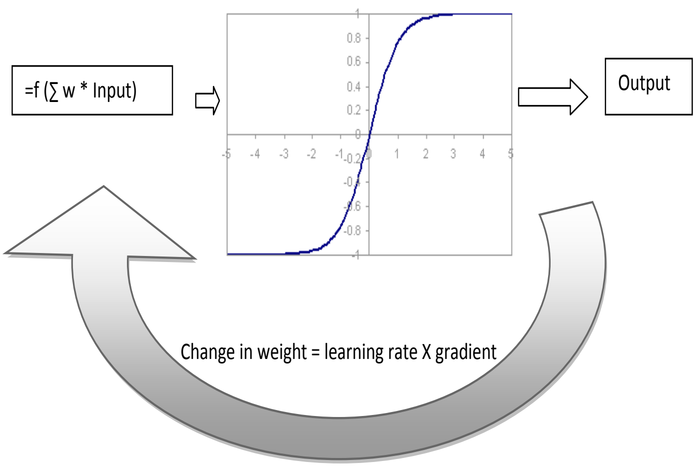

Fig.1 A single node example

The two method deterministic and probabilistic approaches can optimize a function. We use deterministic supervised learning methods as they tend to give better approximation. We want to predict output at time t which is described by the variable yt. We try to find the set of weights that minimise the square of the error. Usually an energy function is described by a single variable such as the Mean Square Error (MSE). When the errors are small the error behaves linearly. In addition, Patternson (1996) proved that the convergence behaviour of the algorithm depends on initial weights and learning rate. In error surface the direction and distance of the optimum can be accurately estimated from the first and second order derivatives. Quadratic function in f(x) can be expressed in matrix notation as follows in Eq. (2).

$$f(x)=XTAx+2bTX+c \tag{2}$$

f(x) achieves its minimum at x = A-1 b, f(x) achieves its maximum at x = - A-1 b, if A is negative definite. All derivative based optimization techniques employ the concept of the Taylor series expansion. In order to train the network, the data set is divided into training and test sets. In order to reduce network over fitting we should test randomly selected data. Finlay (2003) used the error from the networks as stopping parameter for algorithms to determine if training should be stopped when the validation error becomes larger than the training error, the training can be stopped. The way to avoid local minima is by using randomly selected starting points for the weights being optimised. The bias is a measure of how much the network output data sets differs from the desired function. There are techniques for maximizing the generalization of backpropagation algorithm.

3.2 Support vector machine

The airfoil geometry is generated with an innovative inverse design technique based on support vector machine. Support Vector Machine was introduced by Cortes and Vapnik(1997). The SVM model has high prediction accuracy (Ahmet Emin Kurtoglu,2018). Support vector machine methods are based on results of Statistical Learning theory. The SVM is trained using a profile database and used to obtain the geometry of new profile with specified operating conditions and performance parameters. The conformal mapping is applied to design airfoil in early days. Practical application was hampered severely by the lengthy calculations involved in obtaining the airfoil. Mangler and Lighthill showed for the first time that the velocity distribution specified around the airfoil could not be entirely arbitrary. Specifically, the velocity distribution had to satisfy two important integral constraints: 1. guarantee compatibility with the free stream velocity and 2. Ensure closure of the airfoil profile. These theories did much to the inverse approach. All important constraints are expressed in terms of the velocity distribution around the airfoil. The Eppler method allows the airfoil to be divided into a number of segments along each of which the velocity distribution is prescribed with the angle of attack. This approach has matured into a very powerful tool for design.

4. Methodology

4.1 MFNN Model

Neural networks offer a very powerful framework for representing nonlinear mappings from several input variables to several output variables. The system should produce good predictions for

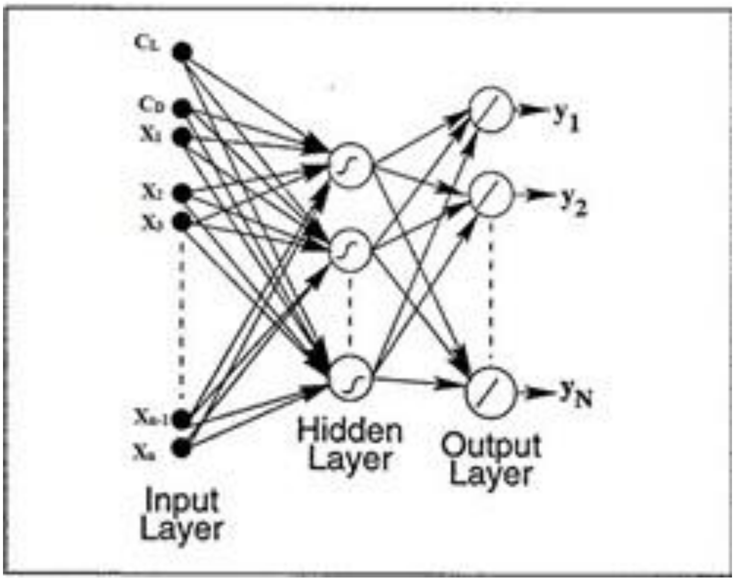

Fig. 2 Neural network trained to predict surface Y-coordinates of Airfoil

new data. Training involves minimization of an appropriate error function. Riedmiller and Braun (1993) said that Learning algorithms such as the back-propagation algorithm for feed-forward multilayer networks help us to find a set of weights by improvement from an arbitrary starting point. An airfoil can be described by a set of x- and y-coordinates.

The aerodynamic coefficients corresponding to series of airfoil are stored in a database along with the airfoil coordinates. A feed forward neural network is created with input as an aerodynamic coefficient and the output as the airfoil coordinates. This is then trained to predict the corresponding y-coordinates. The Fig. 2 shows the Neural-Network Model trained to predict Y-coordinates and the coefficients Cl,Cd and X Coordinates are the inputs .

4.2 SVR Model

Simulation is the feasible method because analytical techniques become too difficult to handle. The benchmarks consist of several hundreds of billions of dynamically executed instructions. One needs alternative methods to predict the performance measures the method will use machine learning techniques such as MFNN and SVR which have shown big success in the solution of learning problems. To find optimal parameters, minimizing root mean squared error can be used for determining the optimum function value. Grid search can use a cross validation process, In cross validation the original dataset is partitioned into k subsets. Fig. 3 shows the flowchart of our SVR model for a single fold.

The design space decomposition, where by the boundaries of regions are defined by the variables. The designs as acceptable and unacceptable and explicitly defining the boundaries. The approach does not approximate responses. The explicit boundaries are obtained using an SVR classifier, Cristianini(2002) and Alpaydin(2004).The definition of explicit nonlinear boundaries in a multidimensional space and can form disjoint regions. A set of N training points xi is associated with one of two classes characterized by a value yi . A general expression of the SVR is in Eq. (3)

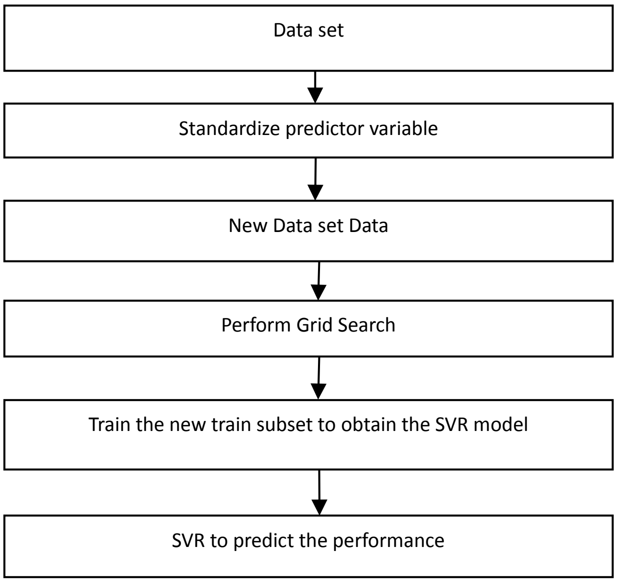

Fig. 3 SVR to optimize model parameters

$$s = b + \sum_{i=1}^{N} \lambda_i y_i k(x_i, x) \tag{3}$$

The construction of explicit boundaries in terms of deterministic variables. It is used to train the SVR consists of classifying the samples into stable and unstable configurations. This classification is performed in two ways. The approach assumes the availability of its approximation for a spectral analysis. Consider a system governed by a set of n first order differential equations:

f X; p

where ‘p’ is a vector of parameters. The n Eigen values i of the Jacobian J of component Jij @fi=@Xj provide the information on the stability of the system. Each sample Xi of the DOE is then assigned a value yi for the training of the SVR.

4.3 Airfoil problem

The problem consists of a 2D airfoil. The stiffness in pitch and plunge are assumed to be polynomials leading to the following Eq. (4) form for the restoring moment M and force Fh:

$$M_{\propto=} = k_{\propto}(\propto + k_{3\propto}\propto^3 + k_{5\propto}\propto^5) \tag{4}$$

$$F_{h=} = k_h(\propto + k_{3h}\propto^3 + k_{5h}\propto^5) \tag{5}$$

where h is the non dimensional plunge. This is a properties of geometric nonlinearities for actual wings make LCOs. Here the terms k3, k5, k3h, and k5h provide the nonlinear terms. In the nonlinear case, Lee *et al.* (1999) derived the equations for cubic stiffness. The integration was performed using the explicit Euler method. The code was parameterized with the following Table 1 quantities:

Table 1 Configuration of Airfoil for the construction of boundaries

Parameter	Value
Reduced velocity	6.0
Mass ratio	40
Elastic-axis midchord separation ah	2.0
Initial pitch	0.05
Initial plunge	0.0
Center-of-mass elastic-axis separation x	1.8
Radius of gyration r	1.86
Damping in pitch and plunge	0
Nonlinear stiffness	0

5. Results and discussion

Evaluation criteria also include the number of parameters as these affect the training time and a small number of parameters give better generalisation ability; one example that does this is the Akaike information criterion (AIC) in Eq .(6), Akaike (1974).

$$AIC(K) = \frac{LLF}{N} + \frac{K}{N} \tag{6}$$

The best input dimension for NN structure be selected using the AIC criterion. When simple models are generated using AIC criterion, it provides better generalisation ability. During training, the performance criteria to be minimised is usually represented by the Mean Square Error (MSE) in Eq. (7).

$$E_k = \frac{1}{2}(T_K - O_{ok})^2 \tag{7}$$

The best AIC is achieved by the networks HONNs. The time taken for training has a direct correlation with the number of weights. The best model has the lowest MSE and training time. The compared convergence curves of MSE's in training process for each machine learning method are given in Fig. 4. The performance of the prediction models are summarized in Table.2, which shows the MSE for the different model. Figure-4 shows how the training decreases MSE with the epoch. The green dotted line indicates the error in MFANN. And the blue line indicates that of error in SVR approach. From the fig. 4, it is obvious that our approach converges quickly compared to other approach. Also the error is minimal for SVR model.

It is said that SVR is one of the fastest and accurate learning algorithms. Our proposed SVR is faster than Levenberg-Marquardt algorithm in MFANN. From the below Table 2 observation, you can see the time taken to converge to solve design problem is less when compared to the other algorithm. In general, the standard algorithm does not perform as well on pattern recognition problems as it does on function approximation problems. The advantage of the back probation algorithm decreases as the number of network parameters increases.

Higher lambda favours gradient descent, lower lambda favours Newton. Cell in this matrix represents the second order derivative of the output of the neural network. The calculation of the Hessian matrix value will be accomplished by calculating the gradients.

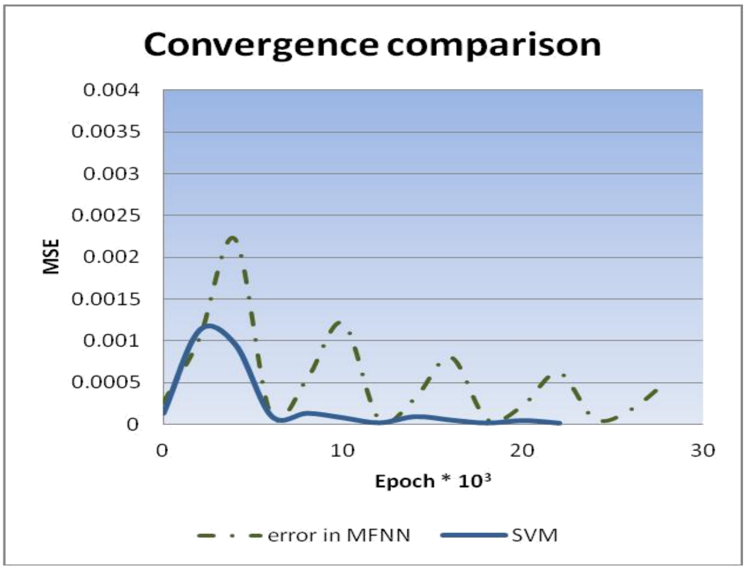

Fig.4 Convergence comparison

Table 2 Performance comparison

SVR			MFNN		
epoch	MSE	Time (ms)	epoch	MSE	Time (ms)
0	0.108857	16	1	0.500698	109
480	0.124079	31	2	0.442961	111
840	0.124255	47	4	0.347800	114
1140	0.123867	62	6	0.314990	116
1500	0.106499	78	8	0.269975	119
1860	0.021690	82	10	0.192142	122
2280	0.000039	94	12	0.098765	125

$$\frac{\partial E}{\partial w_{(i)}} = 2(y-t)\frac{\partial y}{\partial w_{(i)}}$$

The LMA algorithm only supports a single output. LMA algorithm has its roots in mathematical function approximation concepts. The Levenberg-Marquardt algorithm has some drawback; the each epoch will take more time to complete the computation. In the Table 2 the Levenberg-Marquardt algorithm take more time to complete each epoch. In our investigation of neural network models for inverse design of airfoil sections, we found that satisfactory results were obtained by using the SVR model.

In Table 3, we have given the values of stored y coordinates, the values of calculated y coordinates (calculated using our algorithm) for a pattern NACA1017 and also the difference

Table 3 Profile comparison between calculated and stored Y coordinates

Y coordinate database	Y coordinate calculated using proposed algorithm in test phase	Difference
0.00350	0.003086	0.000414
0.00993	0.009320	0.000610
0.03173	0.030728	0.001002
0.04468	0.044099	0.000581
0.05721	0.056115	0.001095
0.07540	0.074828	0.000572
0.07835	0.078136	0.000214
0.07593	0.075784	0.000146

Table 4 Maximum error for airfoils

Airfoil	Maximum error (%)
NACA0012	0.00194
NACA0014	0.00265
NACA2013	0.00113
NACA2017	0.00238

between these values. We have given some sample coordinates out of 26 coordinates for the pattern NACA1017. From the table 3, we can say that the computed profiles generated during the SVR process show good agreement with the database profiles.

We computed the error by considering the y coordinate we derived from our approach as yi(computed) and the y coordinate from the database as yi(actual) and tabulated in Table 4 and this table shows the maximum error in percentage for the airfoils shown in Fig. 5. From this table, we can conclude that the SVR predicated comparatively the correct airfoil profiles.

The new computed airfoil x and y coordinates from the SVR process are passed to XFoil tool. This tool creates the airfoil and generates its corresponding CL, CD. Four such airfoils are shown in Figure 5.

In Table 5 the results obtained are compared and tabulated. When you check the table, you can easily find that our proposed algorithm converges quickly than the other approaches. This table clearly proves that the SVR approach results in less error and takes less time to predict the airfoil for the given CL, CD.

The following observations could be gained from the results:

- In general, the results show that the SVR method performs much better than MFANN methods.
- The SVR model yields the lowest error (MSE = 0.000016) for the prediction of Airfoil design.
- The MFANN model yields the lowest error (MSE = 0.000103) for the prediction of Airfoil design.
- Execution times of the SVR prediction models within 94ms.

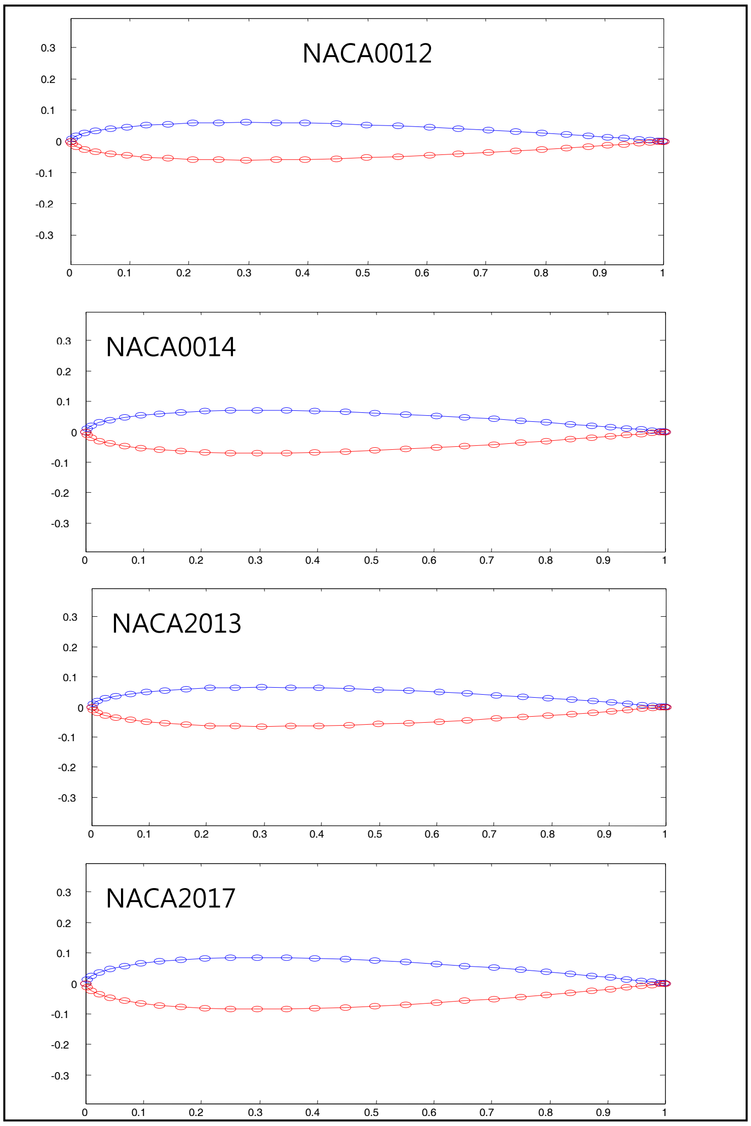

Fig. 5 Generated airfoil profiles

6. Conclusions

In advances in machine learning, the design of Airfoil becomes a realistic, high lift and low drag promises to deliver the important performance. The performance measures of the 2D Airfoil design is an important task and simulation is a time consuming process for this operation. We developed data driven SVR and MFNN models to predict the 2D airfoil architecture. The nonlinear aero elastic problems, which relies on the construction of explicit LCO boundaries using SVR. These can design the airfoil accurately and fast without time consuming. Among the

Table 5 Comparison table

Epoch * 1000	error in MFNN approach	SVR Model
0	3.20248	3.011418
2	0.00013	0.000131
4	0.001444	0.001125
6	0.003933	0.000961
8	0.00014	0.000101
10	0.001281	0.000041
12	0.002752	0.000141
14	0.000065	0.000061
16	0.00020	0.000021
18	0.001336	0.000093
20	0.000294	0.000056
22	0.000106	0.000018
24	0.000926	0.000068
26	0.000499	0.000016
28	0.000039	Converged and the a attained
30	0.000202	-
32	0.000064	-
34	0.000024	-
36	0.000063	-
38	0.000294	-
40	0.000034	-
42	0.000538	-
44	0.000096	-
46	0.000172	-
48	0.000245	-
50	0.000067	-
52	0.000151	-
54	0.000187	-
56	0.000051	-
58	0.000138	-
60	0.000136	-
62	0.000039	-
64	0.000134	-
66	0.000099	-
68	0.000037	-
70	0.00012	-
72	0.000095	-
74	0.000041	-
76	0.000098	-

models, the SVR model shows the best performance for airfoil design. In future the research can include more number of variables and the use of multi fidelity approaches to reduce computational expenses.

References

Abid, F.F. and Najim, M. (2001), "A fast feedforward training algorithm using a modified form of the standard backpropagation algorithm", *IEEE Trans. Neural Networks*, **12**, 424-430.

Allen, M. and Maute, K. (2004), "Reliability-Based Design Optimization of Aeroelastic Structures", *Struct. Multidisciplinary Optimization*, 27(4), 228-242. https://doi.org/10.1007/s00158-004-0384-1.

Alpaydin, E. (2004), *Introduction to Machine Learning*, MIT Press, Cambridge, USA.

Kurtoglu, A.E. (2018), "Patch load resistance of longitudinally stiffened webs: Modeling via support vector machines", *Steel Compos. Struct.,* **29**(3).

Cristianini, N. and Shawe-Taylor, J. (2000), *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press.

Dowell, E. and Tang, D. (2002), "Nonlinear aero elasticity and unsteady aerodynamics", *AIAA J.*, **40**(9), 1697-1707. https://doi.org/10.2514/2.1853.

Finlay, D.D., Nugent, C.D., McCullagh, P.J., Black, N.D. and Lopez, J.A. (2003), "Evaluation of a statistical prediction model used in the design of neural network based ECG classifiers: A multiple linear regression approach", 258-26.

Garabedian, P.R. and Korn, D.G. (1971), "Numerical design of transonic airfoils", *Numerical Solution of Differential Equations - II*, Academic Press, Elsevier, Germany.

Genbrugged, D. and Eeckhout, L. (2007), "Statistical simulation of chip multiprocessors running multi-program workloads", *Proceedings of 25th International Conference on Computer Design*, IEEE, 464-471.

Han, F., Ling, Q.H. and D.S. Huang (2008), "Modified constrained learning algorithms incorporating additional functional constraints into neural networks"*, Information Sci.*, **178**, 907-919.

Hertz, J., Krogh, A. and Palmer, R.G., (1991), *Introduction to the Theory of Neural Computation*, Addison-Wesley Publishing Co., California, USA.

Jeong, S.Y. and Lee, S.Y. (2000), "Adaptive learning algorithms to incorporate additional functional constraints into neural networks", *Neuro Comput.*, **35**,73-90.

Fan, J. (2014), "Accelerating the modified Levenberg-Marquardt method for nonlinear equations", *Math. Comp*., 1173-1187.

Kousen, K. and Bendiksen, O. (1994), "Limit cycle phenomena in computational transonic aeroelasticity," *J. Aircraft*, **31**(6), 1257-1263. https://doi.org/10.2514/3.46644.

Lee, B., Jiang, L. and Wong, Y. (1998), "Flutter of an Airfoil with Cubic Restoring Force, Journal of Fluids and Structures", **13**(1), 75-101. https://doi.org/10.1006/jfls.1998.0190.

Maute, K., Nikbay, M. and Farhat, C. (2003), "Sensitivity analysis and design optimization of three-dimensional nonlinear aeroelastic systems by the adjoint method", *J. Numerical Methods Eng*., **56**(6), 911-933.

Patternson, D.W. (1996), *Artificial Neural Network: Theory and Applications*, Prentice-Hall, Englewood Cliffs, NJ, USA.

Pettit, C. (2004), "Uncertainty quantification in aeroelasticity—Recent results and research challenges", *J. Aircraft*, **41**(5), 1217-1229. https://doi.org/10.2514/1.3961.

Pires, J.C.M., Martins, F.G., Sousa, S.I.V. and Alvim, M.C.M. (2008), "Pereira M.C. Selection and validation of parameters in multiple linear and principal component regressions", *Environ. Modelling Software*, **23**(1), 50-55. https://doi.org/10.1016/j.envsoft.2007.04.012.

Saltan, M. and Terzi, S. (2008), "Modeling deflection basin using artificial neural networks with cross-validation technique in backcalculating flexible pavement layer moduli", *Adv. Eng. Software*, **39**(7), 588-592, https://doi.org/10.1016/j.advengsoft.2007.06.002.

Sejnowski, T. and Rosenberg, C. (1987), "Parallel networks that learn to pronounce English text", *Complex Syst.*, **1**, 143-168.

Dutta, S., Samui, P. and Kim, D. (2018), "Comparison of machine learning techniques to predict compressive strength of concrete", *Comput. Concrete*, **21**(4), 463-470. https://doi.org/10.12989/cac.2018.21.4.463.

Zayid, E.I.M. and Akay, M.F. (2012), "Predicting the performance measures of a message-passing multiprocessor architecture using artificial neural networks", *Neural Comput. Appl.*, **23**(7-8), 2481-2491. https://doi.or/10.1007/s00521-012-1267-9.

TK