11th ANKARA INTERNATIONAL AEROSPACE CONFERENCE AIAC-2021-102
8-10 September 2021 - METU, Ankara TURKEY

GENERATION OF AERODYNAMIC DATABASE MODELS USING MACHINE LEARNING ALGORITHMS WITH CLASSICAL AND ADAPTIVE SAMPLING METHODS

Ertan Demiral[1] and Çağatay Şahin[2]
ROKETSAN
Ankara, Turkey

ABSTRACT

Generation of accurate aerodynamic database models for the flight vehicles, which can be fulfilled by performing high fidelity CFD (Computational Fluid Dynamics) analyzes, is of importance to carry out realistic computer-based simulations before the flight tests. The number of CFD analyzes to be performed is usually limited due to high computational costs. This limitation often challenges the creation of proper aerodynamic database models and hence requires the use of efficient data sampling methods and machine learning algorithms. In this study, different sampling methods and machine learning algorithms are utilized and compared with each other in terms of nrmse (normalized root mean square error) metric. Results indicate that, adaptive sampling method, which decides the points at which CFD analyzes are performed, can reduce required number of analyzes up to 30% by achieving similar model accuracy compared to classical sampling methods. Apart from that, data modeling studies show that lower error metric values can be obtained with Gaussian process based algorithms compared to standard neural network algorithm.

INTRODUCTION

In this study, it is aimed to compare data sampling methods and machine learning algorithms for the generation of aerodynamic database models of conventional Tandem Control Missile (TCM). Representative figure of the TCM configuration is given below:

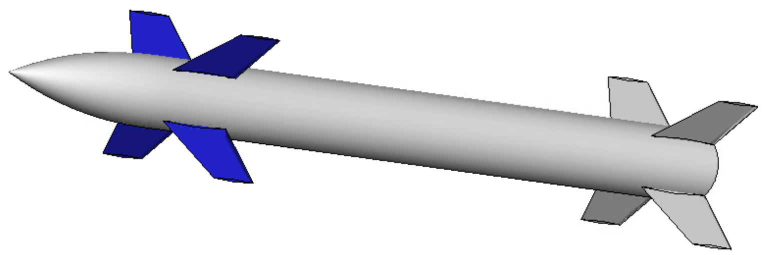

Figure 1: TCM configuration

[1] Aerodynamic Engineer at ROKETSAN, Email: ertan.demiral@roketsan.com.tr
[2] Aerodynamic Engineer at ROKETSAN, Email: cagatay.sahin@roketsan.com.tr

Aerodynamic database model of this flight vehicle (TCM configuration) in the scope of this work composes of 6 inputs (features) and 6 static aerodynamic coefficients (labels), as defined in Table 1. In this model, Mach number, angle of attack and side slip angle input features are used to specify flow properties. On the other hand, pitching, yawing and rolling panel deflection angle input features define the attitude of the blue colored (Figure 1) canard control surfaces. Moreover, labels correspond to force and moment coefficients with respect to body axes coordinate system and are used to calculate forces and moments in dimensional form on the flight vehicle at any flight condition defined with stated inputs (features).

Table 1: Aerodynamic database parameters

			Definition	**Unit**
Features	M	:	Mach Number	-
	α	:	Angle of Attack	°
	β	:	Side Slip Angle	°
	δ_e	:	Pitching Panel Deflection Angle	°
	δ_r	:	Yawing Panel Deflection Angle	°
	δ_a	:	Rolling Panel Deflection Angle	°
Labels	CA	:	Axial Force Coefficient	-
	CY	:	Yaw Force Coefficient	-
	CN	:	Normal Force Coefficient	-
	CLL	:	Rolling Moment Coefficient	-
	CLN	:	Yawing Moment Coefficient	-
	CM	:	Pitching Moment Coefficient	-

METHOD

In this study, four different machine learning algorithms, which are GP (Gaussian Process), DNN (Deep Neural Network), DKL (Deep Kernel Learning) and DSPP (Deep Sigma Point Process) and three different sampling methods which are lhs (latin hypercube sampling), sobol and adoe (adaptive design of experiment) are employed in order to generate aerodynamic database model for the TCM configuration. Implementations have been achieved making use of “pytorch”, “gpytorch”, “ax-platform”, “botorch” and “openturns” open source python libraries. Details of the corresponding algorithms are explained in the following sections.

Machine Learning Algorithms

<u>GP (Gaussian Process)</u>

Gaussian Process (GP) is a powerful machine learning method, which is well known for regression problems. This method’s prediction f(x) is not an exact value, but rather a probability distribution ($f(x) \sim N(\mu(x), \sigma^2(x))$) over all admissible functions that fit the data. GP is specified by a mean function m(x) which corresponds to mean value at any point of the input space and a covariance function $K(x,x')$ that sets the covariance between points. Working principle of GP is based on assuming a prior in terms of mean and kernel functions, calculating the posterior with training data by optimizing hyperparameters of kernel (covariance) function with a gradient or stochastic based optimizers and computing the predictive distribution on points of interest.

DNN (Deep Neural Network)
Neural networks are computing systems, which are inspired by densely connected human brain cells (neurons). The structure, which is formed by computing neurons and connections between them (weights), is given in Figure 2. They are used for predicting output variable as a function of the inputs (independent variables). Neural networks learn things by processes called feed forward and backpropagation. They involve predicting outputs, comparing them with the true outputs, and using the difference between them to modify the weights of the connections between the neurons in the network. By increasing neuron numbers in hidden layers, models with high complexity can be generated. Therefore, they are very beneficial for learning from large datasets.

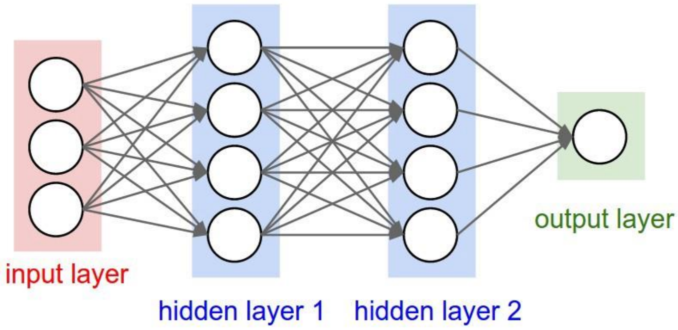

Figure 2: Representative neural network structure

DKL (Deep Kernel Learning)
Deep kernel learning is a Gaussian process that incorporates neural networks to improve the performance of the model with feature extraction. In short, it is a serial combination of two learning systems. Specifically, it transforms the inputs of a spectral mixture base kernel with a deep architecture. Then the properties of these kernels are jointly learned through the marginal likelihood of a Gaussian process.

DSPP (Deep Sigma Point Process)
DGP (Deep Gaussian Process) is a natural generalization of GPs in which a sequence of GP layers form a hierarchical model, like deep neural networks with GP neurons, in which the outputs of one GP layer become the inputs of the subsequent layer, resulting in a flexible, compositional function prior. DSPP (Deep Sigma Point Process) is enhanced version of DGP, replacing the continuous mixture of Normal distributions of DGP with a finite one which is learned during the training process. Model structures are given in Figure 3.

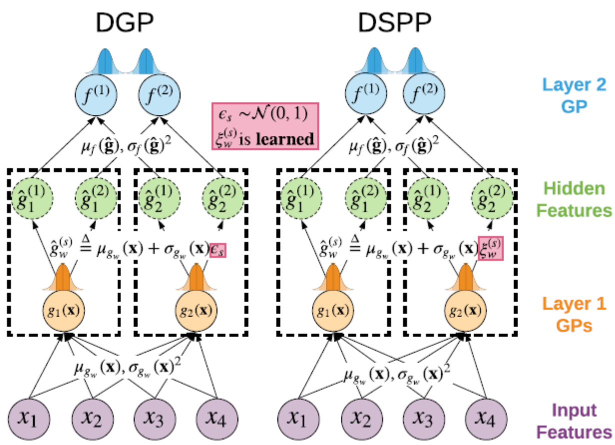

Figure 3: Model structures of DGP (Deep Gaussian Process) and DSPP (Deep Sigma Point Process) [Jankowiak, M., Pleiss, G., & Gardner, J. (2020)]

Sampling Algorithms

In many engineering problems, the total number of function evaluations is drastically limited by computational budget; hence, it is of crucial interest to develop methods for efficiently selecting candidate designs for the experiment. This artificial input space filling process is named as DoE (Design of Experiments). There are two main strategies for sampling input space, classical and adaptive DoE.

Classical DoE

The classical DoE strategies determine all candidate design points without any result consideration and then compute all responses independently. This approach relies on the idea of filling input space homogeneously or with a prescribed sequence. In this study, lhs (latin hypercube sampling) and sobol which are the most commonly used sampling methods in the literature are considered.

Adaptive DoE

Adaptive approach enables adapting the experimental design based on the past observations. This approach builds the DoE sequentially, by choosing a new candidate point as a function of the previously determined points and their corresponding response values. In this study, NIPV (Negative Integrated Posterior Variance) acquisition function that decide a new design point which decreases global uncertainty at most is utilized to achieve adaptive experimentation. Formulation of this function is given below (BOTORCH: A Framework for Efficient Monte-Carlo Bayesian Optimization [Balandat, M., Karrer, B., & Jiang, R., J., Daulton, S., Letham, B., Wilson, A., G., Bakshy, E. (2020)]):

$$\mathrm{NIPV}(\mathbf{x}) = -\int_{\mathbb{X}} \mathbb{E}\big[\mathrm{Var}(f(x)\,|\,\mathcal{D}_{\mathbf{x}}) \mid \mathcal{D}\big]\,dx.$$

Adaptive experimentation is implemented based on the flow chart in Figure 4 where it is seen then iterative design suggestion continues until both one design for each label is queried at one iteration and it is reached to one hundred iterations.

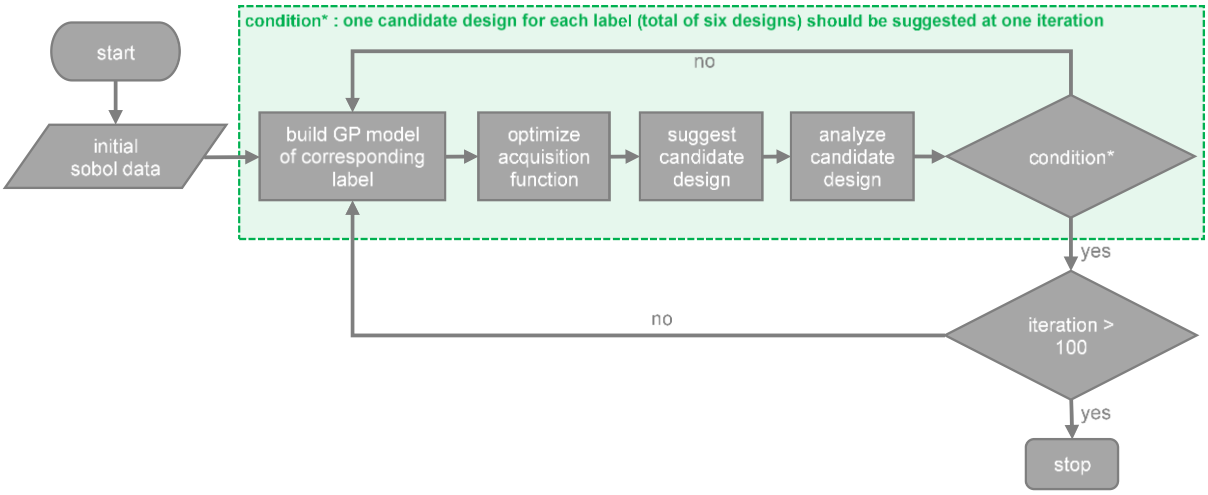

Figure 4: Adaptive experimentation flow chart

In above flow chart, candidate designs are analyzed using Missile DATCOM aerodynamic fast prediction tool instead of CFD tool, since it is computationally affordable and the purpose is to make a comparative study.

RESULTS

Classical and adaptive experimentations have been carried out at three different Mach regimes, subsonic (0.1< Mach < 0.7), transonic (0.7< Mach < 1.3) and supersonic (1.3< Mach < 3.0). Results of the experiments have been presented as nrmse (normalized root mean square error) metric with respect to iteration number in Figure 5, 6 and 7.

Iteration number at the x-axes of the plots can be considered as generated total number of design points. At each iteration, total of six design points, one for each label variable, are proposed sequentially by the NIPV acquisition function or sobol/lhs algoirthms and then these designs are analyzed to obtain corresponding label values. Hence, total of six hundred design points are proposed at the end of the one hundred iteration.

The performance of sampling methods have been measured using nrmse metric (lower is better) of which formulation is given in Equation 1. Metric values have been calculated at each iteration by building/updating a proper GP model and then making predictions on a particular test data with this model.

$$\boldsymbol{rmse = root\ mean\ square\ error}$$
$$\boldsymbol{nrmse = normalized\ root\ mean\ square\ error}$$

$$\boldsymbol{nrmse = \frac{rmse}{average\ of\ absolute\ values\ of\ label\ test\ data} \times 100 \quad (Equation\ 1)}$$

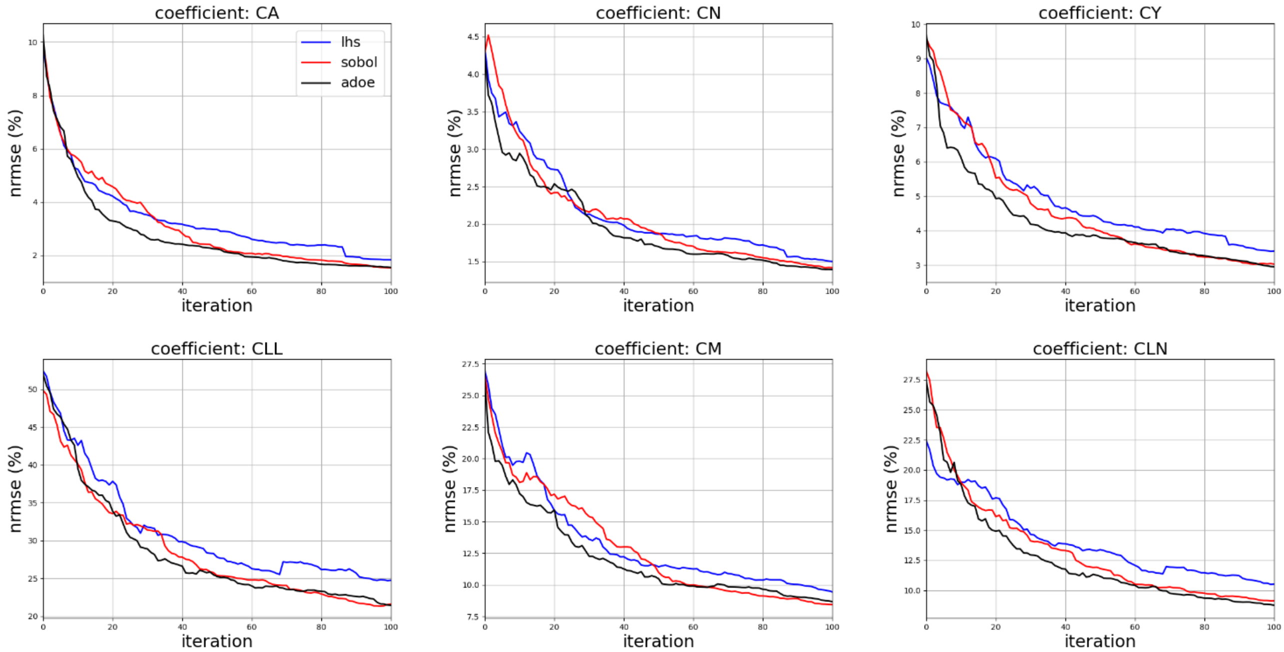

Figure 5: nrmse (normalized root mean square error) history for each coefficient at subsonic (0.1 < Mach < 0.7) speed regime

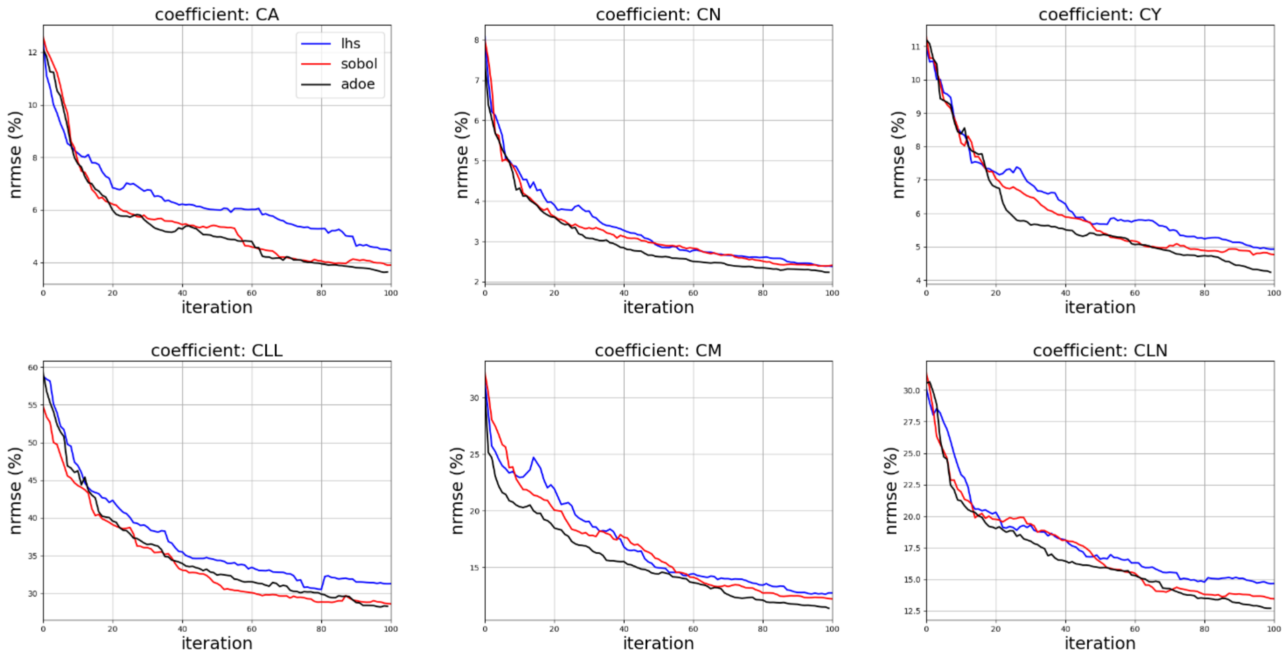

Figure 6: nrmse (normalized root mean square error) history for each coefficient at transonic (0.7 < Mach < 1.3) speed regime

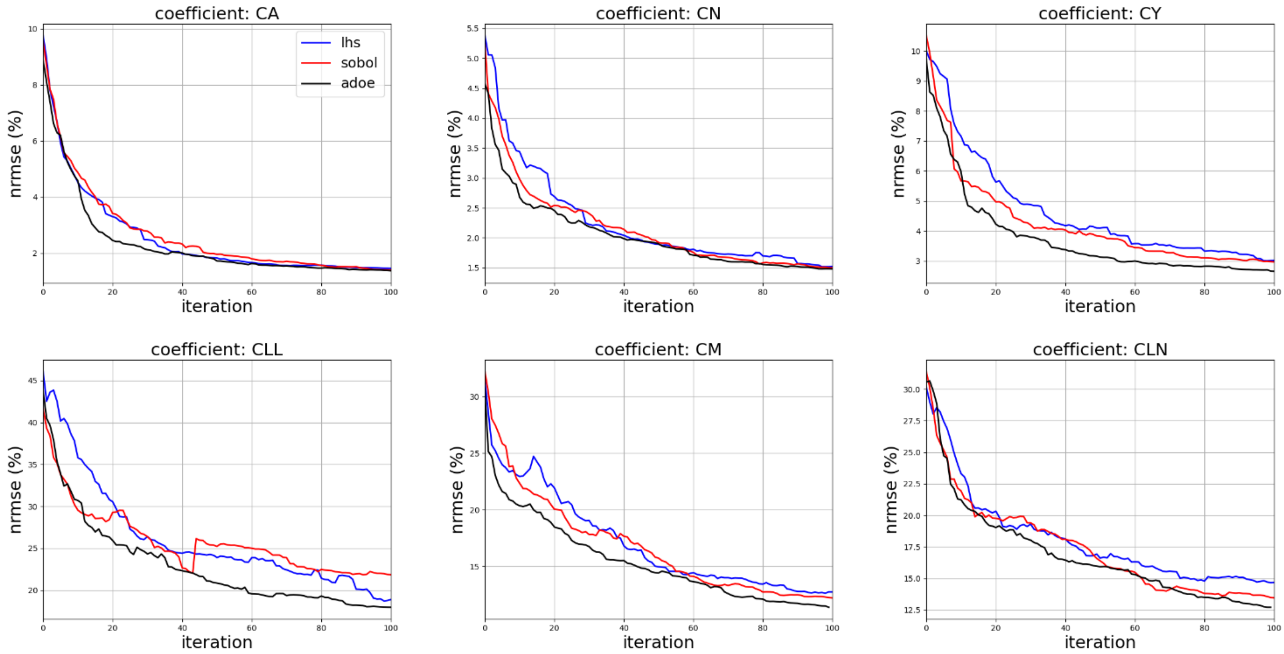

Figure 7: nrmse (normalized root mean square error) history for each coefficient at supersonic (1.3 < Mach < 3.0) speed regime

In above figures, results of the experiments regarding nrmse metric values of sampling methods (lhs, sobol and adoe) for each label variable (CA, CN, CY, CLL, CM, CLN) are presented. Since the feature space is quite large and experimentations have been performed only one time, nrmse metric shows wiggly behavior. From the results, it is seen that adoe sampling method generally have lower nrmse metric value compared to lhs and sobol sampling methods at each Mach regime.

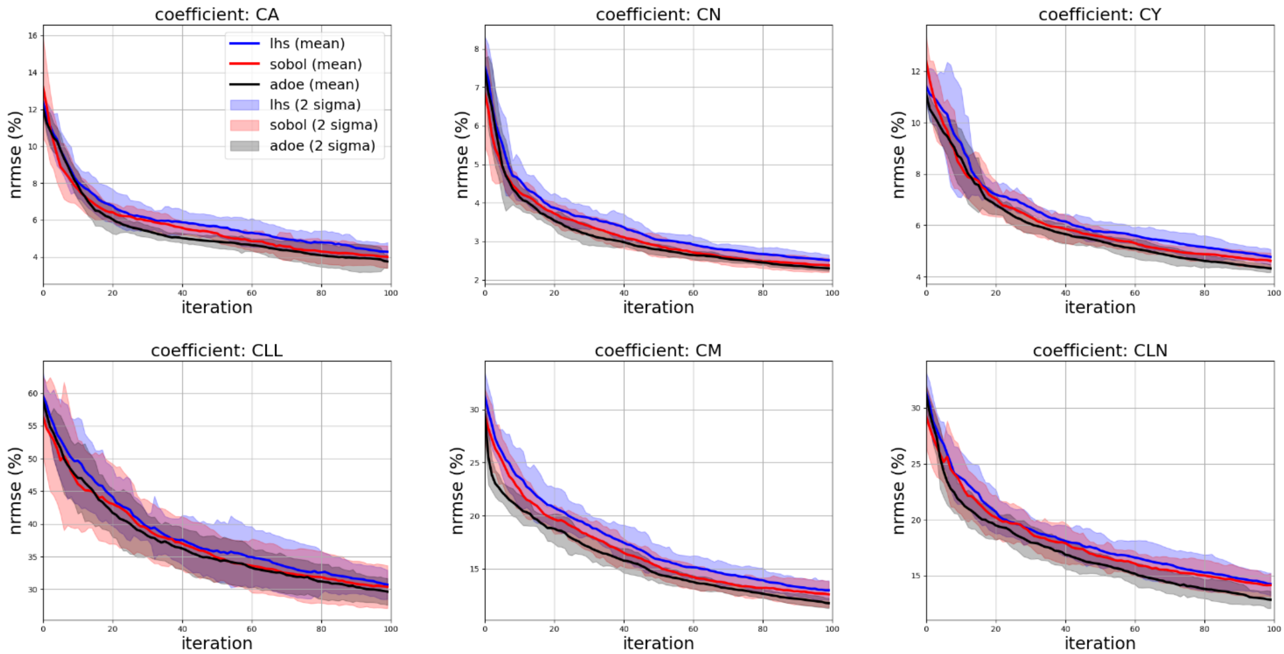

Figure 8: nrmse (normalized root mean square error) history for each coefficient at transonic (0.7 < Mach < 1.3) speed regime with 10 times repeated

In addition, experimentation at transonic mach regime is repeated ten times for each sampling method to obtain better interpretable results than it is shown in Figure 6, 7 and 8. The results of repeated experimentation are presented as mean and confidence bound (2 sigma standard deviation) of nrmse metric with respect to iteration number in Figure 8. From figure, it is seen that adoe sampling method has lower nrmse metric values compared to lhs and sobol.

After successful completion of adoe experimentation for all Mach regimes, nearly 1800 data points have been collected and these are used as training data for machine learning algorithms which are driven to obtain optimum aerodynamic database models. Below table indicates performance values of driven algorithms in terms of nrmse metric:

Table 2: Comparison of machine learning algorithms in terms of nrmse metric for each label (values in the cell correspond to nrmse metric values and lower is better)

Method / Labels	DNN	GP	DKL	DSPP
CA	5.10	**3.10**	3.72	3.37
CY	3.91	**3.71**	9.58	3.83
CN	2.73	1.99	2.92	**1.71**
CLL	20.73	22.07	20.90	**20.30**
CLN	12.60	**11.27**	14.48	11.70
CM	13.71	11.56	14.85	**10.36**

From above table, it is seen that DSPP and GP algorithms generally reach lower nrmse metric values for interested label variables compared to DNN and DKL methods.

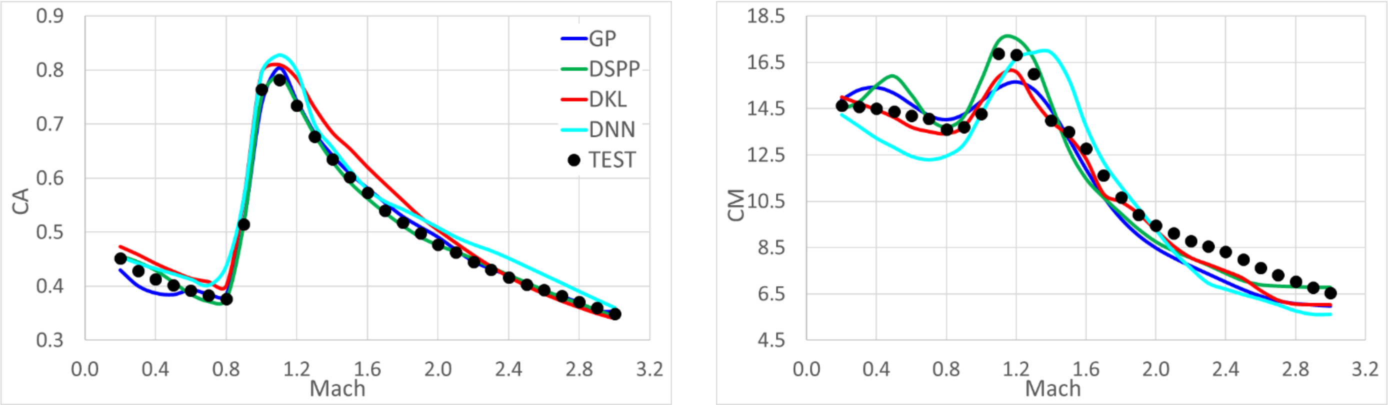

Figure 9: Mach sweep comparisons of model predictions for CA and CM labels with test data

($\alpha = 10°$, $\beta = 0°$, $\delta_e = 10°$, $\delta_r = 0°$, $\delta_a = 0°$)

In addition to the tabulated nrmse metric values, mach sweep plots of the stated algorithms for two selected labels, CA and CM, are presented in Figure 9. From figure, it is seen that machine learning algorithms are capable of capturing nonlinear behaviors which is generally seen around the transonic speed regimes.

CONCLUSION

In this paper, a comprehensive study regarding data sampling methods and data driven machine learning algorithms has been achieved to generate more accurate aerodynamic database models for flight vehicles. According to sampling results, it is seen that adaptive sampling algorithm (adaptive design of experiment) can generally outperform classical sampling methods in terms of nrmse (normalized root mean square error) metric at all interested Mach regimes. This high performance can also be interpreted as similar model accuracies with adaptive samplings can be obtained with less number of analyzes than with classical methods and this is crucial when computational budget is limited. In addition, results of data modelling studies with machine learning algorithms indicate that Gaussian based algorithms are generally more successful than standard neural network based algorithm in terms of nrmse metric.

References

Balandat, M., Karrer B., Jiang, D. R., Daulton, S., Letham, B., Wilson, A. G., Bakshy, E (2020) *BOTORCH: A Framework for Efficient Monte-Carlo Bayesian Optimization*

Gardner, J. R., Pleiss, G., Bindel, D., Weinberger, K. Q., Wilson, A. G. (2019) *GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration*

Jankowiak, M., Pleiss, G., & Gardner, J. (2020). Deep sigma point processes. In Conference on Uncertainty in Artificial Intelligence (pp. 789-798). PMLR.

Picheny, V., Ginsbourger, D., Roustant, O., Haftka, R. T., & Kim, N. H. (2010). Adaptive designs of experiments for accurate approximation of a target region. Journal of Mechanical Design, 132(7).

Wilson, A. G., Hu, Z., Salakhutdinov, R., & Xing, E. P. (2016). Deep kernel learning. In Artificial intelligence and statistics (pp. 370-378). PMLR.