

**Jihad A. Badra<sup>1</sup>**

Transport Technologies Division, R&DC,  
Saudi Aramco,  
Dhahran 31311, Eastern Province, Saudi Arabia  
e-mail: jihad.badra@aramco.com

**Fethi Khaled**

King Abdullah University of Science and  
Technology,  
Clean Combustion Research Center,  
Thuwal 23955, Saudi Arabia  
e-mail: fethi.khaled@kaust.edu.sa

**Meng Tang**

Aramco Services Company,  
Aramco Research Center-Detroit,  
Novi, MI 48377  
e-mail: meng.tang@aramcoamericas.com

**Yuanjiang Pei**

Aramco Services Company,  
Aramco Research Center-Detroit,  
Novi, MI 48377  
e-mail: yuanjiang.pei@aramcoamericas.com

**Janardhan Kodavasal**

Cummins Technical Center,  
Columbus, IN 46237  
e-mail: janardhan.kodavasal@cummins.com

**Pinaki Pal**

Energy Systems Division,  
Argonne National Laboratory,  
Lemont, IL 60439  
e-mail: pal@anl.gov

**Opeoluwa Owoyele**

Energy Systems Division,  
Argonne National Laboratory,  
Lemont, IL 60439  
e-mail: owoyele@anl.gov

**Carsten Fuetterer**

Friendship Systems AG,  
Potsdam 14482, Germany  
e-mail: fuetterer@friendship-systems.com

**Brenner Mattia**

Friendship Systems AG,  
Potsdam 14482, Germany  
e-mail: brenner@friendship-systems.com

**Farooq Aamir**

King Abdullah University of Science and  
Technology,  
Clean Combustion Research Center,  
Thuwal 23955, Saudi Arabia  
e-mail: aamir.farooq@kaust.edu.sa

# Engine Combustion System Optimization Using Computational Fluid Dynamics and Machine Learning: A Methodological Approach

*Gasoline compression ignition (GCI) engines are considered an attractive alternative to traditional spark-ignition and diesel engines. In this work, a Machine Learning-Grid Gradient Ascent (ML-GGA) approach was developed to optimize the performance of internal combustion engines. ML offers a pathway to transform complex physical processes that occur in a combustion engine into compact informational processes. The developed ML-GGA model was compared with a recently developed Machine Learning-Genetic Algorithm (ML-GA). Detailed investigations of optimization solver parameters and variable limit extension were performed in the present ML-GGA model to improve the accuracy and robustness of the optimization process. Detailed descriptions of the different procedures, optimization tools, and criteria that must be followed for a successful output are provided here. The developed ML-GGA approach was used to optimize the operating conditions (case 1) and the piston bowl design (case 2) of a heavy-duty diesel engine running on a gasoline fuel with a research octane number (RON) of 80. The ML-GGA approach yielded >2% improvements in the merit function, compared with the optimum obtained from a thorough computational fluid dynamics (CFD) guided system optimization. The predictions from the ML-GGA approach were validated with engine CFD simulations. This study demonstrates the potential of ML-GGA to significantly reduce the time needed for optimization problems, without loss in accuracy compared with traditional approaches. [DOI: 10.1115/1.4047978]*

*Keywords: machine learning, internal combustion engine, optimization*

## Introduction

The global demand for energy used in the transportation sector is expected to continue rising at an annual rate of 1–1.5% by 2040 according to recent projections [1,2]. This increase is mainly driven by the expected rise in population, gross domestic product

<sup>1</sup>Corresponding author.

Contributed by the Internal Combustion Engine Division of ASME for publication in the JOURNAL OF ENERGY RESOURCES TECHNOLOGY. Manuscript received July 2, 2020; final manuscript received July 7, 2020; published online August 27, 2020. Assoc. Editor: Hameed Metghalchi.

(GDP), and living standards. Currently, internal combustion (IC) engines, fueled by petroleum-derived liquid hydrocarbons (gasoline and diesel), dominate the passenger and commercial transportation sectors with over 99% market share. IC engines are expected to remain the major source of the transportation energy demand in the interim future, despite significant growth in alternative energy and competing technologies (e.g., electric and fuel cells) [1–3].

The legislative requirements aimed at reducing tailpipe emissions, improving vehicle efficiency, and mitigating the impact of transportation on CO<sub>2</sub> emissions are the main drivers when it comes to changes in the transportation landscape. Those legislative requirements, along with consumer demands for vehicles with improved efficiency, drivability, and affordability, are pushing automakers to explore many aspects of engine designs, combustion control, and after-treatment systems that simultaneously reduce fuel consumption and emissions.

Internal combustion engines have been around since the 19th century, and their conceptual identity as a fuel-powered machine has not changed since. There have been significant technological improvements to their performance in response to fuel efficiency and emission regulations [4]. The tools used to co-optimize the fuel/engine system have evolved over the years. Until 20 years ago, experimental prototyping was the main optimizing method. It was followed by numerical simulations, including complex three-dimensional (3D) computational fluid dynamics (CFD), which played a major role in the engine/fuel system optimization. This development was enabled by the significant advancements in computing power (supercomputers, clusters, and parallelization) and numerical models (turbulence, combustion, spray, heat transfer, meshing, and moving boundaries). Due to high dimensionality, complexity, and highly nonlinear dependencies in engine properties and responses, both experimental and numerical optimization approaches can be inefficient and take a significant amount of time and effort to obtain local rather than global optimum designs and operating conditions [5–7]. To overcome the issues with manual optimizations, alternative approaches have been developed over the years. The design of experiments (DoE) technique is commonly used now to enhance design optimization [8]. Probst et al. [8] showed that the sequential DoE approach, coupled with CFD models, can successfully and efficiently optimize engines. The sequential DoE optimization was compared with an optimization performed using a genetic algorithm (GA). Their study highlighted the strengths of both methods for optimization. The GA (known to be an efficient and effective method) found a better optimum, while the DoE method found a good optimum with fewer total simulations. The DoE method also ran more simulations concurrently, which is an advantage when sufficient computing resources are available. GAs [9–14] have also been used to facilitate design optimization, where an objective function is defined to represent CFD simulations. This GA approach often yields better optimum solutions compared with DoE-based optimization.

Significant advancements have been made in the area of artificial intelligence (AI) in the last couple of decades. These advances have enabled machine learning (ML) to be a potentially effective tool to optimize engine/fuel systems [15–18]. Particularly, neural networks have been used in several studies to efficiently optimize engine/fuel systems. The real-time engine control system correction [19,20] and the effects of fuel properties on engine emissions [7,21] are examples of some recent applications. Other ML algorithms, such as random forest and support vector machines, have also been used in engine and vehicle-related problems [22–24]. Recently, Moiz et al. [25] proposed an improved Machine Learning-Genetic Algorithm (ML-GA) approach based on an ensemble machine learning technique known as SuperLearner [26], to optimize the operating conditions of a heavy-duty (HD) gasoline compression ignition (GCI) engine. DoE was first performed to generate a database to train an ML algorithm. The ML optimization algorithm was then used without any further CFD modeling to search for optimum engine design parameters. Their approach yielded comparable optimum operating conditions in significantly less computing

time compared with the CFD-GA approach [25]. Other studies that implemented ML-GA algorithms for engine optimization may be found in the literature [7,10–12].

Compression ignition engine technologies have been considered as attractive alternatives because they have the potential to combine the best of gasoline and diesel engines. The CI combustion mode eliminates traditional engine knock observed in spark-ignition (SI) engines, and the globally lean operation reduces the engine-out emissions. Additionally, the lean, throttle-less operation, coupled with higher compression ratios, improves engine efficiency. Better mixing of fuel and air prior to combustion (due to the higher volatility of gasoline) reduces soot production relative to the conventional mixing-controlled (diesel) combustion. Our research group in Saudi Aramco extensively investigated the combustion of various fuels in GCI engines [27–39].

In this work, a Machine Learning-Grid Gradient Ascent (ML-GGA) approach, with similar functionality to that of Moiz et al. [25], is proposed. The developed ML-GGA code builds upon the work of Moiz et al. [25]. In addition, the repeatability of the optimization method is investigated. Also, the validity of the SuperLearner when applied with modest extrapolations to the original design space is investigated. Finally, guidelines are provided to assess solution robustness. The functionality of the ML-GGA code is then extended using a feedback strategy and employed to optimize the piston bowl geometry of an HD-GCI engine.

## Machine Learning-Genetic Algorithm Approach

**Definition and Details.** The ML-GA is an optimization technique that can be used to efficiently optimize internal combustion engines. More details about this approach can be found in Moiz et al. [25], and only a brief description of the main features is provided here. In an engine optimization task, the optimization problem is first posed and the different engine design parameters are defined. A sample of cases with different combinations of these parameters are solved using detailed CFD simulations or are prototyped to define the effect of these design variables on the performance of the studied engine. The performance of the engine is normally measured by a merit function that combines a certain set of output parameters which need to be defined before starting the optimization process. These sets of input variables and output parameters are utilized in a supervised ML algorithm for training purposes. The trained ML routine is then coupled with an optimization tool—in this case a GA—to search for the optimum set of engine design variables that lead to improved performance. Moiz et al. [25] used the ML package SuperLearner [26], which is available as an add-on package to the statistical software R [40]. The super-learning technique calculates the optimal combination of a pool of prediction algorithms, such as the Arithmetic Mean (SL.mean), Lasso and Elastic-Net Regularized Generalized Linear Models (SL.gmlnet), Breiman and Cutler's Random Forests for Classification and Regression (SL.randomForest), Support Vector Machines (SL.svm), Extreme Gradient Boosting (Xgboost), Linear Regression Models (SL.lm), and Feed-Forward Neural Networks and Multinomial Log-Linear Models (SL.nnet). The optimal combination is a set of these multiple models with weighting factors that minimize the cross-validated error. The GA method used to find the optimal design by Moiz et al. [25] is the genetic algorithm “malschains” (memetic algorithms with local search chains) [41]. The implementation in R (Rmalschains) was used in their work. Malschains uses a combination of local and global optimization techniques. The idea behind the algorithm is to apply a local search method on the most promising regions, which are found to have the highest fitness value using a (global) genetic algorithm. The GA in malschains is different from a standard genetic algorithm, where the individuals of the population are subjected to genetic operations simultaneously. The malschains algorithm randomly generates an initial population of individuals. The genetic algorithm then evaluates the merit values (fitness) of these individuals and

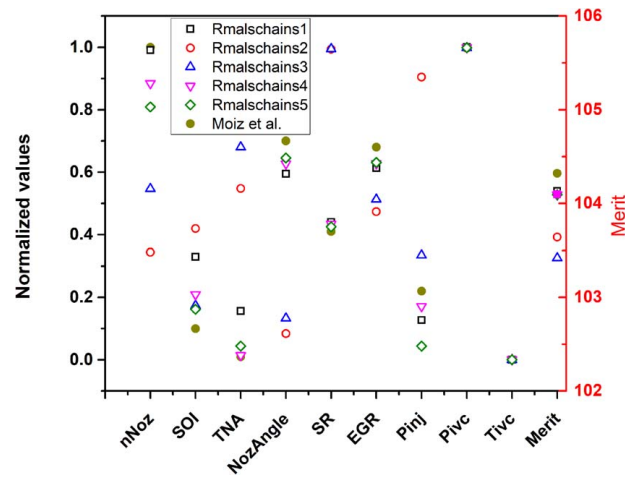
builds a set of individuals that can be further refined by the local search method. The main advantage of the ML-GA technique is the ability to find an optimum design for highly nonlinear problems in a short time for engine optimization tasks, where repeated CFD simulations can be time-consuming and expensive, as detailed in Moiz et al. [25]. They showed that ML-GA reduces the optimization time from about 50 days to 1 day.

**Repeatability and Enhancements in the Machine Learning-Genetic Algorithm Approach.** In this work, we opted to use the ML-GA methodology of Moiz et al. [25] and proposed some modifications/improvements to it. In this section, a detailed investigation of the repeatability ability of the optimization scheme and two additional enhancements are presented and applied to the case study from Moiz et al. [25] to demonstrate their utility. More details about the objective, design variables, and optimization problem may be found in Moiz et al. [25]. In this section, only the impact of the proposed enhancements on the design solution is presented.

**Repeatability.** Although ML-GA presents a resource-efficient and relatively easily implemented tool for engine optimization, it can suffer from repeatability issues where the optimum set of design variables that yields the highest merit are not the same for subsequent runs. This is an anticipated behavior because the Rmalschains optimization method in R is based on random selections, and hence, running it multiple times could lead to different local solutions if not set with care. To avoid finding unrepeatability local optimum designs, special care has to be taken when using certain optimization schemes. To demonstrate this, the ML-GA code of Moiz et al. [25] is reconstructed here. It is then solved using the exact same training data set for five consecutive times while changing the search parameters of the Rmalschains algorithm. Table 1 summarizes the variability among the five best designs found using each set of Rmalschains where the first line (case 0) represents the base parameters used by Moiz et al. [25]. Higher than  $1e^{-4}$  variance means that the merit values of the five best designs are different from each other by more than 1%. It can be clearly seen that all Rmalschains parameters affect the repeatability of the solution, and hence, correctly setting those parameters is critical to ensure reaching the absolute optimum design in terms of merit value. For example, the number of search iterations has to be around 35,000. An iteration number of only 2400 represented

**Table 1 Dependence of the optimal solutions on the parameters' setting of the Rmalschains**

Case	Popsize	ls	Istep	Effort	Alpha	Maxevals	Variance of the five best designs found
0	100	sw	100	0.8	1	35,000	3.83E-07
1	200	cmaes	300	0.5	0.5	2400	1.05E-01
2	100	sw	200	0.8	1	35,000	1.63E-07
3	100	sw	350	0.8	1	35,000	5.05E-07
4	100	sw	3500	0.8	1	35,000	2.23E-07
5	50	sw	3500	0.8	1	35,000	7.76E-07
6	10	sw	3500	0.8	1	35,000	1.44E-04
7	10	cmaes	3500	0.8	1	35,000	2.98E-07
8	10	cmaes	3500	0.8	1	3500	7.29E-04
9	10	cmaes	3500	0.8	1	500	7.01E-02
10	10	cmaes	3500	0.5	1	3500	1.30E-03
11	10	cmaes	3500	0	1	3500	7.58E-03
12	10	cmaes	3500	0	0.5	3500	1.34E-04
13	10	cmaes	3500	0	0.2	3500	8.22E-04
14	10	cmaes	3500	0	0	3500	8.73E-02
15	10	cmaes	3500	0.5	0	3500	1.88E-03
16	10	cmaes	3500	0.5	0	3500	1.15E-01

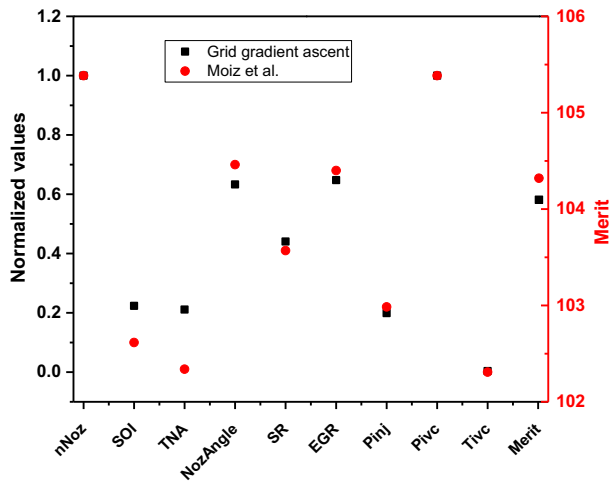


**Fig. 1 Optimum designs obtained using Rmalschains. The five designs obtained from repeating Rmalschains five times are compared with the optimum design parameters reported in Moiz et al. [25].**

by case 1 in Table 1 leads to five different local optimum designs. For illustration purposes, the optimum solutions for case 1 as well as that from Moiz et al. [25] are presented in Fig. 1.

As seen in Fig. 1, the five optimal designs obtained here are different from each other despite having similar merit value (right y-axis). The observed differences can be significant for some design variables such as the number of nozzles (nNoz), start of injection (SOI), total nozzle area (TNA), nozzle angle (NozAngle), swirl ratio (SR), exhaust gas recirculation (EGR), and injection pressure (Pinj). Only the obtained optimum intake valve closing pressure (Pivc) and temperature (Tivc) are similar for the different runs. Finding the best set of optimum solutions using Rmalschains can be a tedious exercise that requires a know-how expertise. As a remedy for this, a new method is adopted in this work by utilizing a different GA, GGA which, although classical and forward, does not include many setup knobs and easy to implement with no black-boxed information. The GGA divides the multidimensional domain ( $d$ ) of the design parameters into equal multidimensional cubes where the axis of each parameter is divided into  $n$  equal segments. The center of each of the  $n^d$  cubes is then used as the initial design for the optimization technique that uses the classical gradient ascent method. At the end of this step,  $n^d$  local optimum designs are obtained and the best among them is chosen as the global optimum design. Applying the ML-GGA for the case study of Moiz et al. [25] leads to an optimal design that has similar merit value as the design reported by their study, as shown in Fig. 2. This repeatable optimum design was obtained by setting the value of  $n$  to 2 which requires  $2^9 = 512$  iterations as compared with 35,000 iterations for the case of Rmalschains. The optimum designs from five runs are the same as the one shown in Fig. 2.

**Extension of Variable Domain.** The optimum designs reported by Moiz et al. [25] and the best designs obtained in this work have some design variables residing on or near the boundaries set by the predetermined limits. These variables are the number of nozzles, temperature at injection valve closing, and TNA. Extending the design parameter range outside their initial preset limits could potentially lead to better performance. To demonstrate this, the ML-GGA algorithm using Moiz et al.'s case study is repeated by allowing a 10% extension of the design parameters around their upper and lower limits. Only a 10% margin is used here so that the predictability of the ML is still acceptable and within the feasible design space. The five best optimum designs are presented in Table 2 in absolute values, and their normalized values are shown in Fig. 3.



**Fig. 2 Optimum design by using the ML-GA versus ML-GGA method. The ML-GA design is taken from Moiz et al. [25].**

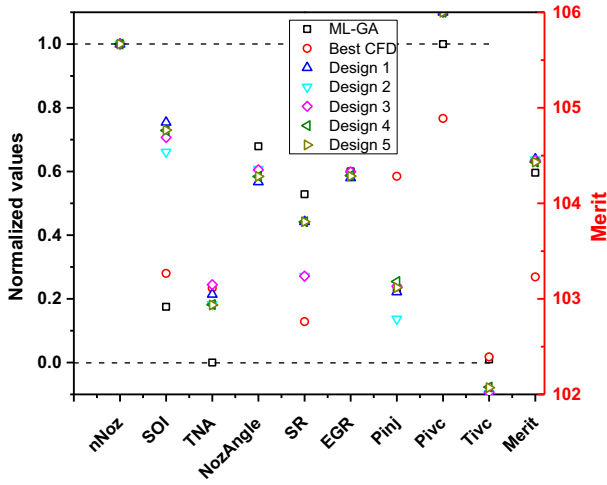
The actual outputs from CFD simulation for the five best designs obtained in this work are also included in Table 2. This CFD verification exercise was performed using the same CFD model employed in Moiz et al. [25]. For reference, we have also included the predicted ISFC and merit value by the ML where good predictability (within less than 0.5%) is achieved. The best ML-GA and CFD-GA results from Moiz et al. [25] are also listed in Table 2. As can be seen from Fig. 3 and Table 2, all optimum designs (1–5) have slightly higher merit values than that of Moiz et al. [25] with extended ranges. In fact, design 1 has an indicated specific fuel consumption (ISFC) that is 0.14% better than the best reported by Moiz et al. [25]. Similarly, the engine-out parameters such as emissions, maximum pressure rise rate (MPRR), and peak cylinder pressure (PCP) are better from the ML-GGA and CFD<sub>ML-GGA</sub> compared with those reported in Moiz et al. [25]. Around 10% reductions in soot and nitric oxides (NO<sub>x</sub>) were obtained from the new simulations. The design parameters that were at their predetermined limits, such as Tivc and Pivc, are now residing at the edge of the

extended ranges (−0.1 and −1.1), suggesting that further extensions are potentially favorable to obtain higher merit values. However, more than 10% extension of the training data space could lead to poor prediction capability of the ML.

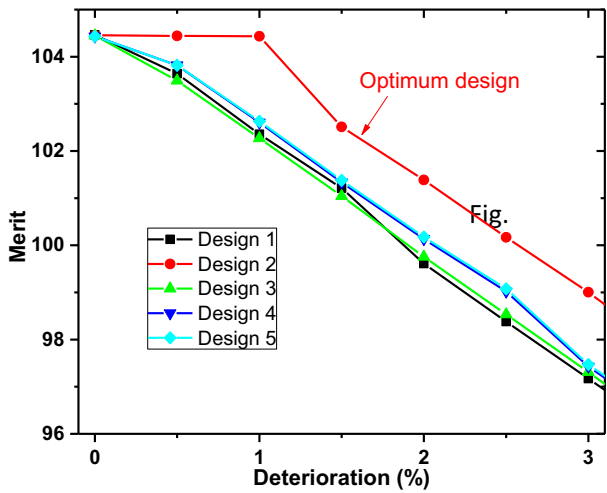
**Post-Processing and Robustness.** An additional enhancement of the ML-GGA technique is to take the advantage of the ML algorithm for the fast qualification of the optimal design in terms of robustness. In many cases, the optimum design is not unique and the optimization process produces multiple optimum designs with similar merit values. The challenge then is to further assess these results and eventually come up with a selection criterion. For engines, the main criterion that a proposed design should offer is robustness. A robust design is a design that will not rapidly deteriorate and lose its merit value once its design parameters experience small perturbations. Multiple reasons can cause perturbations in the design variables. Experimental uncertainties are the main factors for variations in the design variables. Intentional perturbations to the design variables to prevent engine-out parameters such as MPRR, PCP, soot, and NO<sub>x</sub> emissions from hitting their design or regulated limits can be another factor. To prevent the engine performance from significantly deteriorating, a robustness parameter is defined here to evaluate the sensitivity of the merit value on the design parameters. First, the permitted perturbation ranges of the design variables are defined. Subsequently, a new optimization problem can be initiated where the objective is to minimize the merit function, such that the search space is confined around the optimum design within some perturbation radius. The top five local optimum designs, reported in Table 2, are first saved. Afterwards, a sensitivity analysis is used to evaluate the robustness of each of the designs. The sensitivity is performed by looking for the worst design in the vicinity of each local optimum design. The vicinity space is defined as the multidimensional sphere centered around the local optimum design and with a radius equal to the % deterioration rate. Here, we used the Rmalschans to discover the vicinity space of each of the five best designs. The utilization of a Grid Gradient descent scheme in the vicinity of each of the best designs would also be possible. The comparison is shown in Fig. 4. It is clear that design 2 is more robust than the other designs.

**Table 2 Optimum design properties from CFD, Ref. [25], and this work**

Parameter	This work					Reference [25]			
	Design 1	Design 2	Design 3	Design 4	Design 5	ML-GA	CFD <sub>ML-GA</sub>	CFD-GA	
nNoz	10	10	10	10	10	10	10	10	
SOI	−7.98	−8.36	−8.17	−8.09	−8.08	−10.65	−10.65	−10.3	
TNA	1.06	1.05	1.07	1.05	1.05	1.05	1.05	1.0	
NozzleAngle	156.74	157.54	157.5	157.08	157.1	159.26	159.26	158.0	
SR	−1.78	−2.02	−2.02	−1.78	−1.78	−1.81	−1.81	−1.66	
EGR	0.44	0.44	0.44	0.44	0.44	0.45	0.45	0.44	
Pinj	1488.70	1454.73	1496.02	1501.81	1494.11	1492.5	1492.5	1490	
Pivc	2.33	2.33	2.33	2.33	2.33	2.3	2.3	2.3	
Tivc	318.37	318.13	318.34	319.15	319.05	323.0	323.0	323.5	
Engine-out parameters									
Soot (g/(kW h))	ML-GGA	0.0066	0.0082	0.0071	0.0069	0.0070	0.011	0.02	0.022
	CFD <sub>ML-GGA</sub>	0.0082	0.0097	0.0087	0.0084	0.0086			
NO <sub>x</sub> (g/(kW h))	ML-GGA	1.33	1.32	1.34	1.33	1.33	1.32	1.23	1.28
	CFD <sub>ML-GGA</sub>	1.14	1.2	1.22	1.16	1.16			
MPRR (bar/CAD)	ML-GGA	10.32	10.30	10.67	10.36	10.30	13.28	12.22	11.31
	CFD <sub>ML-GGA</sub>	10.16	10.46	11.06	10.25	10.10			
PCP (bar)	ML-GGA	157.08	157.27	158.21	157.41	157.21	166.73	165.23	162.03
	CFD <sub>ML-GGA</sub>	156.78	157.18	157.90	157.20	156.97			
ISFC (g/(kW h))	ML-GGA	153.16	153.18	153.18	153.21	153.21	153.37	153.97	153.85
	CFD <sub>ML-GGA</sub>	153.10	153.09	153.09	153.18	153.17			
Merit	ML-GGA	104.46	104.46	104.45	104.43	104.43	104.32	103.91	104.0
	CFD <sub>ML-GGA</sub>	104.51	104.51	104.52	104.45	104.46			



**Fig. 3** Optimum designs from the ML-GGA method with extended limits. The ML-GA design and best CFD adapted from Moiz et al. [25] are also shown.

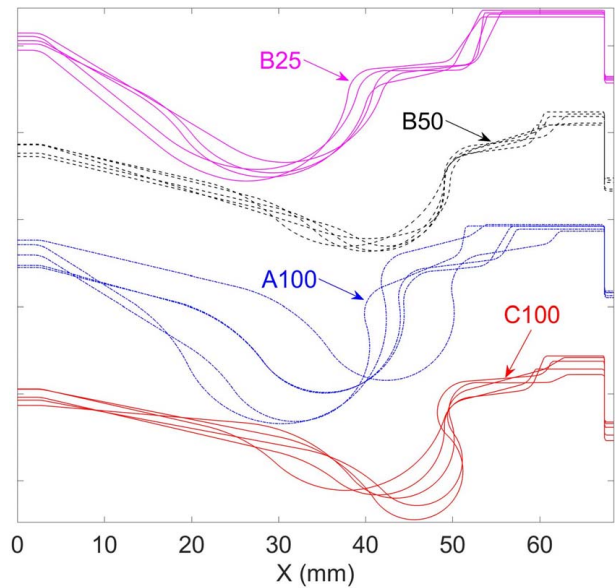


**Fig. 4** Robustness of the five local optimum designs

### Piston Bowl Optimization

**Problem Definition.** A different engine application is studied here to test the capability of the developed ML-GGA model. The optimization of the piston bowl geometry of an HD-GCI engine is performed here. Pei et al. [39] conducted a CFD guided combustion system optimization study of a heavy-duty diesel engine running on a gasoline fuel with a research octane number (RON) of 80. A two-stage DoE approach was adopted in their work. The first stage focused on the piston bowl shape optimization, and the second addressed refinement of the combustion recipe. For optimizing the piston bowl geometry, a software tool, CAESES, was utilized to automatically perturb key bowl design parameters.

CAESES is a highly specialized computer-aided design (CAD) system for the parametric design and optimization of flow-exposed geometries, such as various components of internal combustion engines (e.g., ports, manifolds, combustion chambers, and cooling passages). Its CAD modeling approach is focused specifically on a robust and efficient parametric variation of the geometry within an automated process. Models are built up in such a way that inside a pre-defined design space, the generation of infeasible variants is prevented, be it from a pure geometry regeneration point of view, or with regard to set design constraints. In addition to the modeling environment, CAESES also contains tools for process



**Fig. 5** Top five CFD designs at the four operating conditions (adapted from Ref. [39])

automation, i.e., connection to external simulation codes, as well as DoE and optimization algorithms for driving the process. The design object of this study is an axisymmetric piston bowl. Hence, the geometry can be generated simply as a surface of revolution from a parameterized profile. More details about the design generation approach, the constraints, and the design variable limits may be found in Ref. [39].

A set of 256 piston bowl designs were generated by CAESES using a Sobol sequence and were evaluated at four engine operating conditions using multidimensional CFD simulations. Different optimum designs were identified based on the CFD simulations, and these are shown in Fig. 5 for the four studied engine operating points. Further details about the engine, its operating points, optimization techniques, and results may be found in Ref. [39]; only a brief description is provided here. It may be seen that the overall shape of the top five designs at each operating point share some key features (bowl depth, bowl width, and center height), but they are different from each other. This indicates that all the design parameters are affecting the performance of the engine, but some more than others. Subsequently, it will be almost impossible to understand the relationships between inputs and outputs depending on visual inspection, and therefore, ML can be a very useful optimization tool for such a complicated problem.

### Methodology

**Design Variables.** The purpose of the current work is to extend the CFD study conducted by Pei et al. [39] using the ML-GGA methodology developed here to obtain a better design compared with their best CFD design.

First, the design variables were identified. The piston geometry was parametrized using 11 different design variables, as shown in Fig. 6. These 11 variables were allowed to change within their pre-determined ranges (see Table 5 in Ref. [39]). A constraint on the compression ratio (CR) was imposed. The bowlRadius was set as a dependent parameter for achieving the exact compression ratio (16.5). Its value was determined by an internal optimization loop that is triggered once all other parameter values were set (see Fig. 5 in Ref. [39]).

The ML-GGA algorithm developed here was modified to be able to optimize piston bowl geometries instead of engine operating parameters. The modified code can reproduce the piston bowl

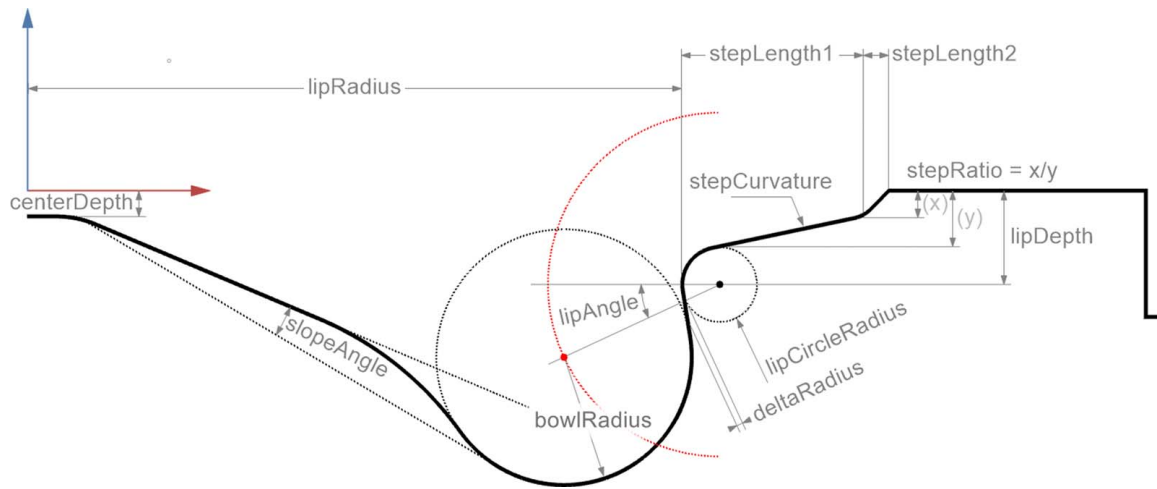


Fig. 6 Piston bowl profile with design parameters

profile for any combination of design variables while maintaining the CR at 16.5. To obey the compression ratio constraint, the volume above the baseline piston bowl profile at the top dead center position is first calculated by the code. For other profiles to have the same compression ratio, it is mandatory that they result in the same clearance volume at the top dead center. For each new piston bowl profile, the clearance volume is first calculated for the bowlRadius of the baseline design. If the clearance volume is different from the baseline value, then the bowlRadius is changed in the direction that minimizes this difference [39]. If no possible bowlRadius able to match the baseline clearance volume, then the suggested design must be discarded. In the Jacobian optimization scheme, this is achieved by attributing a very low merit value to this design in order to push the search direction away from it.

**Training Data.** In ML, training data is critical and generating this data is a key step toward obtaining an efficient algorithm that is able to model engine performance with sufficient accuracy. The quality of training data is measured by the quantity of information it conveys to a learning algorithm. For instance, training data in which the input design parameters only cover small ranges of the feasible space (training data 2 in Fig. 7) can be described as low quality because they will not teach the algorithm about the performance of engine designs in the larger feasible space. Similarly, if the training data are only covering designs with a limited range of

performance functions (training data 1 in Fig. 7), the algorithm would not learn the effect of each of the design parameters on improving the merit function. The actual behavior of the design is in blue. All training sets contain nine data points. Training data 3 is the most effective in obtaining good predictive capabilities as can be visually seen when comparing the perfect behavior with the predicted behavior.

A number of methods can be applied to select the training data, such as Markov decision trees. These algorithms may be categorized as reinforcement learning algorithms. Their main drawback is that they have to be run in a sequential manner, which limits the use of parallel computing. Other simpler methods may be employed where the effect of output is neglected and only the distribution of inputs is diversified as much as possible. These design of experiments tools are widely used and documented [8,42–44]. In the work of Pei et al. [39], the 256 designs used as training data were generated using a Sobol sequence, which is part of CAESSES.

Many ML algorithms, such as neural networks, work better if the input data are normalized. This is due to the fact that neural network parameters are usually obtained based on gradient calculations. These gradient-based problems could easily become ill-conditioned if the inputs are not normalized and have absolute values that are too different. Ill-conditioning of neural networks and the importance of data normalization have been discussed in detail by many authors [45,46]. Therefore, it is recommended to normalize the training data before feeding it to the ML algorithm. After specifying the ranges of each input parameter design, it is straightforward to normalize the input data using a min–max relation. Normalizing the output data could be tricky, especially if no obvious limits of outputs could be anticipated. In this case, the user has to get as wide as possible range of outputs and normalize them accordingly, although output normalization is not as critical as input normalization.

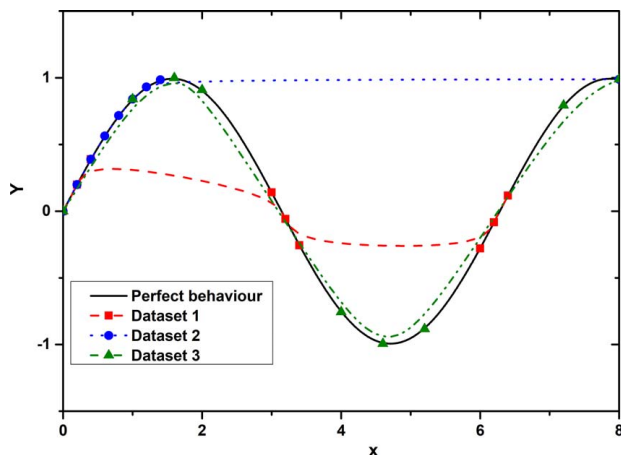
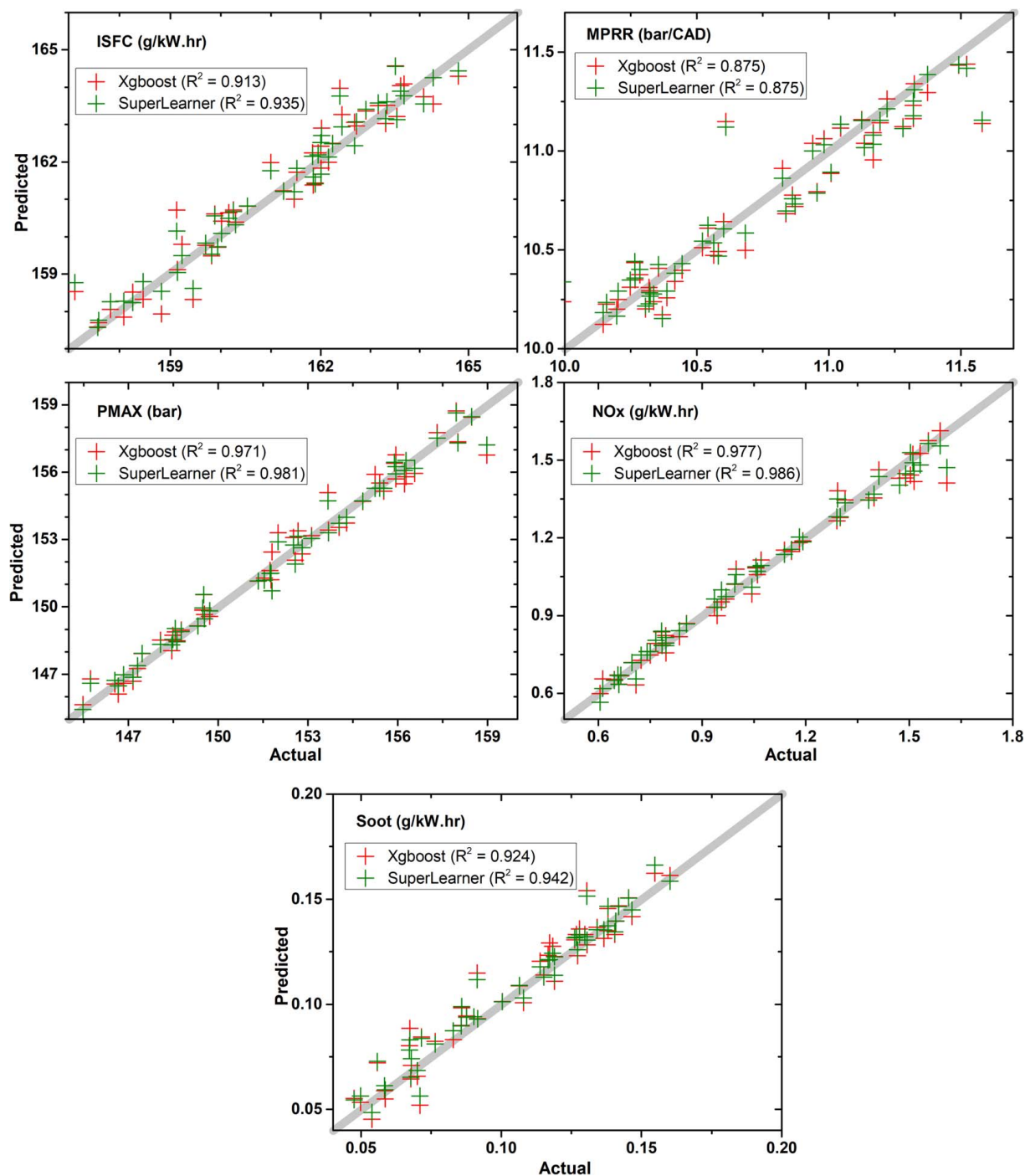


Fig. 7 Effect of training data on predictive capabilities of polynomial fitting

**Fitting and Prediction Capabilities of the Machine Learning Model.** One of the main properties of an unsupervised ML algorithm is its fitting capability. It is important that the ML model is able to fit the performance of the engine design to its design parameters. This fitting capability can be tested on the training data. The ML model should reproduce very well the engine performance of the training data. However, this is not the main objective of the ML model. In fact, the ML model is used to predict the performance of engine designs that are not within the training data, and hence, a better qualification of the ML model is its prediction capability. Usually, there is a compromise between fitting and prediction capability, and hence, care should be taken to avoid over-fitting where the ML model may fit the training data very well but fails to correctly predict new points. In R, a cross-validation (CV) routine is



**Fig. 8 Comparison between predicted and actual CFD results of NO<sub>x</sub> and Soot emissions, fuel consumption (ISFC), maximum pressure (Pmax), and MPRR for 51 testing designs**

available where it is possible to divide the training data into test data and training data and perform a test of the prediction capability of the ML algorithm. Figure 8 presents a comparison between the predicted and actual relevant outputs such as the NO<sub>x</sub>, soot emissions (Soot), ISFC, the maximum pressure (Pmax), and the MPRR of 51 randomly selected test designs when the ML is trained using the remaining 205 designs for the operating case B25 (see Ref. [39] for load conditions). The SuperLearner (that is a combination of many submethods including Xgboost) and Xgboost (only) were tested here.

It is clear from Fig. 8 that SuperLearner performs slightly better than Xgboost. For example, the  $R^2$  of the SuperLearner ISFC

prediction is 0.935 compared with 0.913 of Xgboost. Although the prediction accuracy has some uncertainty, the SuperLearner is able to capture the trend of all relevant outputs for the 51 designs. Increasing the size of the training data would eventually lead to enhancement in the prediction capability, but would also require more CFD simulations.

Figure 9 reports the five-fold cross-validation error in the prediction of ISFC by the ML model, as we increase the size of the training data from 56 to 256 designs. It is noted that the designs chosen in each validation set are randomly selected, and hence, the trend of the cross-validation error is subject to some random error and is presented here for illustration. A firm conclusion about what is the

optimal size of the training data to have minimum prediction error is hard to obtain from Fig. 9, but it is clearly larger than 256. In this work, the final ML model was based on the SuperLearner code, and it was trained using all 256 designs for better prediction accuracy.

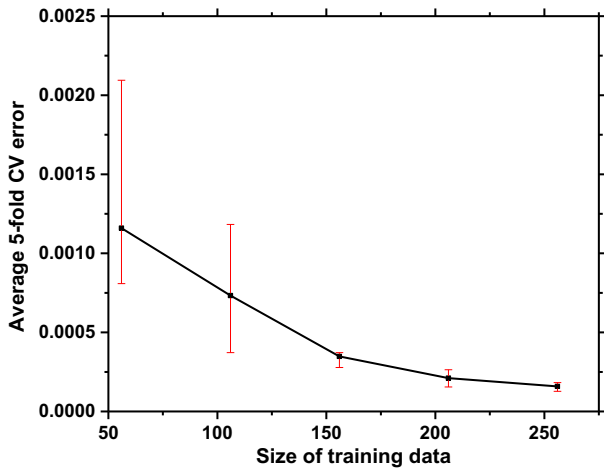
**Merit Function.** The merit function used here is a modification of that used by Pei et al. [39]. Here, output normalization and smoothing (making the merit function differentiable) were taken into account. The merit function is defined as follows:

$$\text{Merit normalized} = 100 - \frac{\left( wf_1 \times \frac{\text{ISFC} - \text{limit}}{\text{limit} - \min(\text{ISFC})} + wf_2 \times f(P_{\max}^*) + wf_3 \times f(\text{MPRR}^*) \right)}{\sum_{i=1}^7 wf_i} - \frac{wf_4 \times f(\text{FSN}^*) + wf_5 \times f(s\text{NOx}^*) + wf_6 \times f(s\text{HC}^*) + wf_7 \times f(s\text{CO}^*)}{\sum_{i=1}^7 wf_i} \quad (1)$$

with

$$f(\text{parameter}^*) = \begin{cases} \frac{\text{parameter} - \text{limit}}{\text{limit} - \min(\text{parameter})}, & \text{if parameter} > \text{limit} \\ 0, & \text{if parameter} \leq \text{limit} \end{cases} \quad (2)$$

where  $wf_i$  and limits of the various parameters are summarized in Table 3.



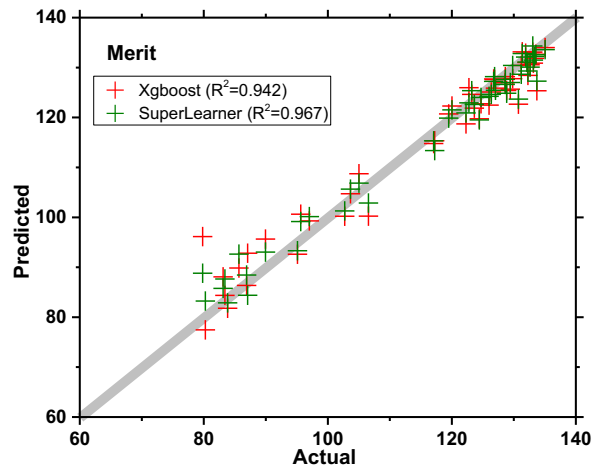
**Fig. 9** Evolution of the cross-validation error of the Superlearner ML a function of the training data size. Five-fold cross-validation error is divided by the number of test designs to get the average five-fold CV error.

**Table 3** Limits and weighting factors used for the merit function calculation

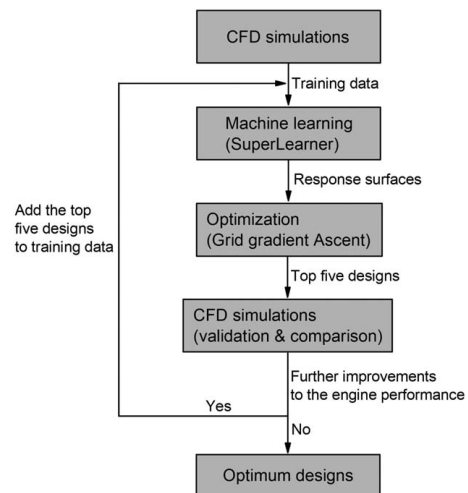
Parameter	Weighting factor ( $wf_i$ )	Limits			
		B25	B50	A100	C100
Pmax (bar)	100			220	
MPRR (bar/CAD)	1			12	
Soot (g/(kW h))	0.1	0.05	0.12	0.1	0.1
NO <sub>x</sub> (g/(kW h))	1	1	1	1.5	1.5
ISFC (g/(kW h))	1			170.7	

The merit function used is the square of the normalized merit defined as follows:

$$\text{Merit} = |\text{Merit normalized}| \times \frac{\text{Merit normalized}}{100} \quad (3)$$



**Fig. 10** Prediction versus actual merit values of the 51 testing designs



**Fig. 11** ML-GGA piston bowl geometry optimization scheme



**Table 4 Design variables, emissions, fuel consumption, merit values, and merit improvements of the best CFD and ML-GGA designs at all operating conditions**

	B25		B50		A100	C100
	Best CFD [39]	ML-GGA	Best CFD [39]	ML-GGA	Best CFD [39]	Best CFD [39]
Piston design variables						
lipAngle	-14.53	-18.59	-7.97	-1.20	11.17	11.17
lipCircleRadius	3.30	3.29	4.16	3.31	3.29	3.29
lipRadius	38.98	38.50	42.27	41.50	43.79	43.79
slopeAngle	2.77	-4.91	-5.20	-8.22	-10.95	-10.95
stepCurvature	0.10	0.07	0.01	0.03	0.02	0.02
stepRatio	0.98	0.78	0.52	0.75	0.76	0.76
stepLength1	12.98	10.17	9.70	10.28	11.70	11.70
stepLength2	2.09	3.66	3.29	3.12	1.89	1.89
centerDepth	2.56	3.77	3.69	4.18	4.41	4.41
bowlRadius	12.00	10.85	12.00	10.78	8.83	8.83
lipDepth	9.77	11.58	8.54	6.49	7.65	7.65
Engine-out parameters						
ISFC (g/(kW h))	161.07	160.54	160.85	160.61	165.76	161.77
Soot (g/(kW h))	0.01	0.01	0.10	0.10	0.05	0.11
NO <sub>x</sub> (g/(kW h))	0.82	0.79	0.99	0.99	1.48	1.49
MPRR (bar/CAD)	9.88	10.18	10.69	10.48	10.03	10.07
Pmax (bar)	97.42	97.19	151.97	151.77	215.69	209.30
Merit	153.29	156.55	137.82	138.79	154.44	142.56
Improvement in merit (%)		2.13		0.70	0.0	0.0

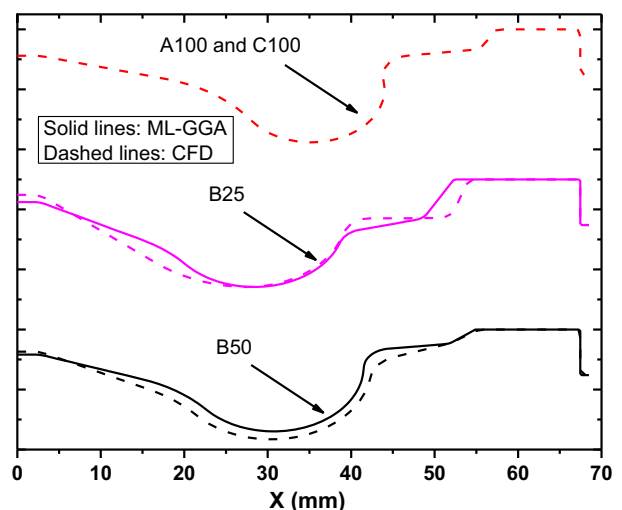
The new merit function definition is consistent with that of Pei et al. [39] resulting in the same order of designs based on the merit values. In our case, the order of the best 30 designs out of the 256 CFD designs are the same for both merit functions (Eq. (3) and [39]). This property is very important to ensure the consistency of the results given by the optimization scheme later.

The same fitting/prediction capability tests were performed for the merit function as well. Figure 10 presents the prediction capability of the merit function using the SuperLearner and Xgboost algorithms. As seen in Fig. 10, the prediction capability of SuperLearner is better than that of Xgboost. This is expected because SuperLearner was shown to predict the engine output parameters better than Xgboost (Fig. 8). In general, the SuperLearner is able to depict the trend of the merit value of engine designs correctly.

**Optimization Scheme.** As seen from Figs. 8 and 10, having only 256 designs for an 11-dimension problem to train the ML algorithm poses a potential issue with predictability. To improve the predictability, an iterative process was implemented here where the ML-GGA was first run using the 256 training data sets. Subsequently, the design variables and the engine output parameters of the five best designs identified by ML-GGA were added to the training data set. The optimum designs identified by the ML-GGA model were solved using CFD before adding them to the training data. This loop was repeated until no further improvements were observed after adding new designs from the previous iteration (see Fig. 11). This process ensures that the predictability of ML around the good designs enhances from one iteration to another. In general, three or four iterations were required to achieve convergence.

**Results.** The ML-GGA optimization scheme described herein was used to identify piston bowl geometries that would yield better engine performance compared with the ones obtained by Pei et al. [39] at the four operating conditions. Better piston bowl designs were obtained from the ML-GGA optimization algorithm at only two operating conditions (B25 and B50). The design variables, engine output parameters, merit values, and improvement in merit for the best CFD and ML-GGA designs are presented in Table 4 for the four operating conditions. The ML-GGA results

listed in Table 4 are from the CFD simulations of the proposed optimum designs by the ML-GGA model. Evidently, 2.13 and 0.7% improvements in merit were obtained by the ML-GGA for B25 and B50 operating points, respectively. The best CFD and ML-GGA piston bowl designs are shown in Fig. 12. The optimum designs obtained from the ML-GGA approach for B25 and B50 share the same high-level features with those from the best CFD case. This is expected because the ML-GGA code essentially refines the search for optimum designs in the vicinity of the best CFD designs. Considering the set of weighting factors derived from the EPA SET Weights shown in Table 6 of Ref. [39], the overall weighted average improvement in merit obtained from the current work is  $2.13\% \times 0.278 + 0.7\% \times 0.278 = 0.78\%$ . Finally, the robustness of the best CFD and ML-GGA designs at all operating conditions is investigated. Figure 13 shows the sensitivity of the merit function on the merit deterioration rate for all optimum designs at the four operating conditions. The deterioration rates are similar for the best CFD and ML-GGA



**Fig. 12 The best designs from CFD [39] and ML-GGA for all operating points**

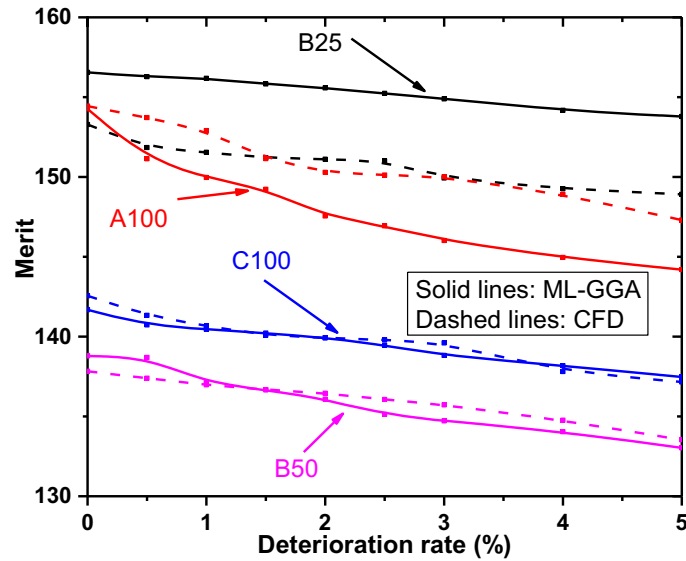


Fig. 13 Robustness of the best designs from CFD [39] and ML-GGA for all operating points

designs at B25, B50, and C100 conditions. The decay of the merit of the ML-GGA design at A100 is, however, higher than that of the best CFD. This indicates that the best CFD at A100 is a more robust design despite having similar merit values to the ML-GGA best designs.

### Conclusions

In this work, a methodical approach for tackling engine optimization problems using machine learning is presented. Critical precautions, recommended algorithms, and suitable optimization techniques are discussed. Internal combustion engine optimization is a highly complex problem because it is coupled and highly non-linear. Therefore, the multidimensional optimization approach can avoid design failure in operation modes beyond the focal optimization criteria. A careful definition of the objective function for optimization is a critical step. Training data for these machine learning algorithms has to be cleverly prepared to improve the prediction efficiency. Global optimum search optimization methods have to be adopted to avoid local optimum designs with reduced merit value. Post-processing of the optimization outputs is also needed to evaluate the recommended design by employing sensitivity and robustness analysis. A Machine Learning-Genetic Algorithm (ML-GGA) code was developed here and validated against literature data. Additional analyses and improvements were introduced to the ML-GA code, and its potential was demonstrated using two case studies. The first case study reproduced and analyzed the recent work performed by Moiz et al. [25], which focused on optimizing the operating conditions of a heavy-duty engine operating in the GCI mode. Our analysis showed that extending the parameter range outside the training data limits and performing a sensitivity study have the potential to yield better optimal conditions. The second case study was aimed at the optimization of the piston bowl geometry of a heavy-duty GCI engine at different operating conditions. Improved piston bowl geometries compared with the best CFD designs were obtained from the ML-GGA code developed here with up to 2.13% improvements in the merit value.

### Acknowledgment

This work has been supported by the Transport Technologies Division at Saudi Aramco R&DC. We would also like to thank Aramco Services Company for their support with the computing cluster at the Aramco Research Center Houston. The submitted

manuscript was created partly by UChicago Argonne, LLC, Operator of Argonne National Laboratory. Argonne, a U.S. Department of Energy (DOE) Office of Science Laboratory, is operated under Contract No. DE-AC02-06CH11357. This research was partly funded by the U.S. DOE Office of Vehicle Technologies, Office of Energy Efficiency and Renewable Energy under Contract No. DE-AC02-06CH11357. Blues High Performance LCRC cluster facilities at Argonne National Laboratory were used for some of the simulations.

The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan. <http://energy.gov/downloads/doe-public-accessplan>

### Conflict of Interest

There are no conflicts of interest.

### Data Availability Statement

The data sets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request. The authors attest that all data for this study are included in the paper. Data provided by a third party are listed in Acknowledgment. No data, models, or code were generated or used for this paper.

### Nomenclature

$p$  = pressure  
 $T$  = temperature

### Subscripts

$inj$  = injection  
 $ivc$  = intake valve closing  
 $max$  = maximum

### References

- [1] ExxonMobil, 2018, "Outlook for Energy: A View to 2040," ExxonMobil, TX, <https://corporate.exxonmobil.com/en/energy/energy-outlook>, Accessed August 9, 2020.
- [2] EIA, 2018, "International Energy Outlook 2018," U.S. Energy Information Administration (EIA), Washington, DC, <https://www.eia.gov/outlooks/ieo/>, Accessed August 9, 2020.

- [3] Kalghatgi, G. T., 2014, "The Outlook for Fuels for Internal Combustion Engines," *Int. J. Engine Res.*, **15**(4), pp. 383–398.
- [4] Kesgin, U., 2004, "Genetic Algorithm and Artificial Neural Network for Engine Optimisation of Efficiency and NO<sub>x</sub> Emission," *Fuel*, **83**(7), pp. 885–895.
- [5] Gen, M., and Cheng, R., 2000, *Genetic Algorithms and Engineering Optimization*, John Wiley & Sons, Hoboken, NJ.
- [6] Manolas, D. A., Frangopoulos, C. A., Gialamas, T. P., and Tsalhis, D. T., 1997, "Operation Optimization of an Industrial Cogeneration System by a Genetic Algorithm," *Energy Convers. Manage.*, **38**(15–17), pp. 1625–1636.
- [7] Wong, K. I., Wong, P. K., Cheung, C. S., and Vong, C. M., 2013, "Modeling and Optimization of Biodiesel Engine Performance Using Advanced Machine Learning Methods," *Energy*, **55**, pp. 519–528.
- [8] Probst, D. M., Senecal, P. K., Chien, P. Z., Xu, M. X., and Leyde, B. P., 2018, "Optimization and Uncertainty Analysis of a Diesel Engine Operating Point Using Computational Fluid Dynamics," *ASME J. Eng. Gas Turbines Power*, **140**(10), p. 102806.
- [9] Zhang, Q., Ogren, R. M., and Kong, S.-C., 2016, "A Comparative Study of Biodiesel Engine Performance Optimization Using Enhanced Hybrid PSO-GA and Basic GA," *Appl. Energy*, **165**, pp. 676–684.
- [10] Wickman, D. D., Senecal, P. K., and Reitz, R. D., 2001, "Diesel Engine Combustion Chamber Geometry Optimization Using Genetic Algorithms and Multi-Dimensional Spray and Combustion Modeling," SAE International, No. 01-0547.
- [11] Hanson, R., Curran, S., Wagner, R., Kokjohn, S., Splitter, D., and Reitz, R. D., 2012, "Piston Bowl Optimization for RCCI Combustion in a Light-Duty Multi-Cylinder Engine," *SAE Int. J. Engines*, **5**(2), pp. 286–299.
- [12] Bertram, A. M., Zhang, Q., and Kong, S.-C., 2016, "A Novel Particle Swarm and Genetic Algorithm Hybrid Method for Diesel Engine Performance Optimization," *Int. J. Engine Res.*, **17**(7), pp. 732–747.
- [13] Shi, Y., and Reitz, R. D., 2010, "Optimization of a Heavy-Duty Compression-Ignition Engine Fueled With Diesel and Gasoline-Like Fuels," *Fuel*, **89**(11), pp. 3416–3430.
- [14] Wu, Z., Rutland, C. J., and Han, Z., 2018, "Numerical Optimization of Natural Gas and Diesel Dual-Fuel Combustion for a Heavy-Duty Engine Operated at a Medium Load," *Int. J. Engine Res.*, **19**(6), pp. 682–696.
- [15] He, Y., and Rutland, C. J., 2002, "Modeling of a Turbocharged DI Diesel Engine Using Artificial Neural Networks," SAE Technical Paper No. 0148-7191.
- [16] He, Y., and Rutland, C. J., 2003, "Neural Cylinder Model and Its Transient Results," SAE Technical Paper No. 0148-7191.
- [17] Samadani, E., Shamekhi, A. H., Behroozi, M. H., and Chini, R., 2009, "A Method for Pre-Calibration of DI Diesel Engine Emissions and Performance Using Neural Network and Multi-Objective Genetic Algorithm," *Iran. J. Chem. Chem. Eng.*, **28**(4), pp. 61–70.
- [18] Vaughan, A., and Bohac, S. V., 2013, "A Cycle-to-Cycle Method to Predict HCCI Combustion Phasing," ASME 2013 Internal Combustion Engine Division Fall Technical Conference, Dearborn, MI, Oct. 13–16, American Society of Mechanical Engineers, p. V001T003A026.
- [19] Krijnsen, H. C., van Leeuwen, J. C. M., Bakker, R., van den Bleek, C. M., and Calis, H. P. A., 2001, "Optimum NO<sub>x</sub> Abatement in Diesel Exhaust Using Inferential Feedforward Reductant Control," *Fuel*, **80**(7), pp. 1001–1008.
- [20] Malikopoulos, A. A., Assanis, D. N., and Papalambros, P. Y., 2009, "Real-Time Self-Learning Optimization of Diesel Engine Calibration," *ASME J. Eng. Gas Turbines Power*, **131**(2), p. 022803.
- [21] de Lucas, A., Durán, A., Carmona, M., and Lapuerta, M., 2001, "Modeling Diesel Particulate Emissions With Neural Networks," *Fuel*, **80**(4), pp. 539–548.
- [22] Orfila, O., Saint Pierre, G., and Messias, M., 2015, "An Android Based Ecodriving Assistance System to Improve Safety and Efficiency of Internal Combustion Engine Passenger Cars," *Transp. Res. Part C: Emerg. Technol.*, **58**(Part D), pp. 772–782.
- [23] Bergmeir, P., Nitsche, C., Nonnast, J., Bargende, M., Antony, P., and Keller, U., 2014, "Using Balanced Random Forests on Load Spectrum Data for Classifying Component Failures of a Hybrid Electric Vehicle Fleet," 2014 13th International Conference on Machine Learning and Applications, Detroit, MI, Dec. 3–6, pp. 397–404.
- [24] Rychetsky, M., Ortmann, S., and Glesner, M., 1999, "Support Vector Approaches for Engine Knock Detection," Proceedings of the International Joint Conference on Neural Networks (IJCNN'99), Washington, DC, July 10–16, Vol. 962, pp. 969–974, Cat. No.99CH36339.
- [25] Moiz, A., Pal, P., Probst, D., and Pei, Y., 2018, "A Machine Learning-Genetic Algorithm (ML-GA) Approach for Rapid Optimization Using High-Performance Computing," SAE Technical Paper No. 01-0190.
- [26] Polley, E. C., and Van Der Laan, M. J., 2010, "Super Learner in Prediction", U.C. Berkeley Division of Biostatistics Working Paper Series, Working Paper 266, <https://biostatistics.berkeley.edu/paper266>
- [27] Viollet, Y., Chang, J., and Kalghatgi, G., 2014, "Compression Ratio and Derived Cetane Number Effects on Gasoline Compression Ignition Engine Running With Naphtha Fuels," *SAE Int. J. Fuels Lubr.*, **7**(2), pp. 412–426.
- [28] Badra, J., Elwardany, A., Sim, J., Viollet, Y., Im, H., and Chang, J., 2016, "Effects of In-Cylinder Mixing on Low Octane Gasoline Compression Ignition Combustion," SAE Technical Paper No. 0148-7191.
- [29] Badra, J. A., Sim, J., Elwardany, A., Jaasim, M., Viollet, Y., Chang, J., Amer, A., and Im, H. G., 2016, "Numerical Simulations of Hollow-Cone Injection and Gasoline Compression Ignition Combustion With Naphtha Fuels," *ASME J. Energy Resour. Technol.*, **138**(5), p. 052202.
- [30] Badra, J., Bakor, R., AlRamadan, A., Almansour, M., Sim, J., Ahmed, A., Viollet, Y., and Chang, J., 2018, "Standardized Gasoline Compression Ignition Fuels Matrix," SAE Technical Paper No. 0148-7191.
- [31] Zhang, Y., Voice, A., Tzanetakis, T., Traver, M., and Cleary, D., 2016, "An Evaluation of Combustion and Emissions Performance With Low Cetane Naphtha Fuels in a Multicylinder Heavy-Duty Diesel Engine," *ASME J. Eng. Gas Turbines Power*, **138**(10), p. 102805.
- [32] Badra, J., Viollet, Y., Elwardany, A., Im, H. G., and Chang, J., 2016, "Physical and Chemical Effects of low Octane Gasoline Fuels on Compression Ignition Combustion," *Appl. Energy*, **183**, pp. 1197–1208.
- [33] Chang, J., Kalghatgi, G., Amer, A., and Viollet, Y., 2012, "Enabling High Efficiency Direct Injection Engine With Naphtha Fuel Through Partially Premixed Charge Compression Ignition Combustion," SAE Technical Paper No. 0148-7191.
- [34] Chang, J., Viollet, Y., Alzubail, A., Abdul-Manan, A. F. N., and Al Arfaj, A., 2015, "Octane-on-Demand as an Enabler for Highly Efficient Spark Ignition Engines and Greenhouse Gas Emissions Improvement," SAE Technical Paper No. 0148-7191.
- [35] Chang, J., Viollet, Y., Amer, A., and Kalghatgi, G., 2013, "Fuel Economy Potential of Partially Premixed Compression Ignition (PPCI) Combustion With Naphtha Fuel," SAE Technical Paper No. 0148-7191.
- [36] Viollet, Y., Abdullah, M., Alhajhouje, A., and Chang, J., 2015, "Characterization of High Efficiency Octane-on-Demand Fuels Requirement in a Modern Spark Ignition Engine With Dual Injection System," SAE Technical Paper No. 0148-7191.
- [37] Atef, N., Badra, J., Jaasim, M., Im, H. G., and Sarathy, S. M., 2018, "Numerical Investigation of Injector Geometry Effects on Fuel Stratification in a GCI Engine," *Fuel*, **214**, pp. 580–589.
- [38] Badra, J. A., Sim, J., Viollet, Y., Zhang, Y., and Chang, J., 2017, "CFD Guided Gasoline Compression Ignition Engine Calibration," ASME 2017 Internal Combustion Engine Division Fall Technical Conference, Seattle, WA, Oct. 15–18, American Society of Mechanical Engineers, p. V002T006A012.
- [39] Pei, Y., Pal, P., Zhang, Y., Traver, M., Cleary, D., Futterer, C., Brenner, M., Probst, D., and Som, S., 2019, "CFD-Guided Combustion System Optimization of a Gasoline Range Fuel in a Heavy-Duty Compression Ignition Engine Using Automatic Piston Geometry Generation and a Supercomputer," SAE Technical Paper No. 01-0001.
- [40] Polley, E., LeDell, E., Kennedy, C., Lendle, S., and van der Laan, M., 2018, "Package 'SuperLearner,'" CRAN, <https://mran.microsoft.com/snapshot/2018-03-03/web/packages/SuperLearner/SuperLearner.pdf>, Accessed August 9, 2020.
- [41] Bergmeir, C. N., Molina Cabrera, D., and Benítez Sánchez, J. M., 2016, "Memetic Algorithms With Local Search Chains in R: The Rmallschains Package," *J. Stat. Soft.*, **75**(4), pp. 1–33.
- [42] Fritz, S., Hötendorfer, H., and Koller, M., 2014, "Design of Experiments in Large Diesel Engine Optimisation," *MTZ Ind.*, **4**(1), pp. 40–45.
- [43] Wilson, V. H., 2012, "Optimization of Diesel Engine Parameters Using Taguchi Method and Design of Evolution," *J. Braz. Soc. Mech. Sci. Eng.*, **34**(4), pp. 423–428.
- [44] Hicks, C. R., and Turner, K. V., 1999, *Fundamental Concepts in the Design of Experiments*, Oxford University Press, New York.
- [45] Sola, J., and Sevilla, J., 1997, "Importance of Input Data Normalization for the Application of Neural Networks to Complex Industrial Problems," *IEEE Trans. Nucl. Sci.*, **44**(3), pp. 1464–1468.
- [46] Saarinne, S., Bramley, R., and Cybenko, G., 1993, "Ill-Conditioning in Neural Network Training Problems," *SIAM J. Sci. Comput.*, **14**(3), pp. 693–714.