Fuzzy-tree-constructed data-efficient modelling methodology for volumetric efficiency of dedicated hybrid engines

Li, Ji; Zhou, Quan; Williams, Huw; Xu, Pu; Xu, Hongming; Lu, Guoxiang

DOI:
10.1016/j.apenergy.2022.118534

Document Version
Peer reviewed version

Citation for published version (Harvard):
Li, J, Zhou, Q, Williams, H, Xu, P, Xu, H & Lu, G 2022, 'Fuzzy-tree-constructed data-efficient modelling methodology for volumetric efficiency of dedicated hybrid engines', *Applied Energy*, vol. 310, 118534. https://doi.org/10.1016/j.apenergy.2022.118534

Fuzzy-Tree-Constructed Data-Efficient Modelling Methodology for Volumetric Efficiency of Dedicated Hybrid Engines

Ji Li[1], Quan Zhou[1], Huw Williams[1], Pu Xu[2], Hongming Xu[1], and Guoxiang Lu[2] ✉

[1]Department of Mechanical Engineering, School of Engineering, University of Birmingham, B15 2TT, UK

[2]BYD Company Limited, Department of New Technology Development, 518118, China

✉ E-mail: lu.guoxiang@byd.com.

Highlights

- A new methodology of fuzzy-tree-constructed data-efficient modelling is proposed.
- Gaussian distributed resampling technique is developed for reducing experimental efforts.
- Experiments at steady engine operations are used to train and validate the fuzzy trees.
- The proposed methodology achieves superior learning efficiency with fewer samples.

Abstract

The accurate characterization of volumetric efficiency is essential for modern combustion engines to achieve better performance, lower emissions, and reduced fuel consumption. To minimize experimental effort on sample collection and maintain high-precision volumetric efficiency characterization, this paper proposes a new methodology of fuzzy-tree-constructed data-efficient modelling to precisely quantify the air mass flow through the engine. Differing from conventional data-driven modelling, this methodology introduces a hierarchical fuzzy inference tree (HFIT) with three original topologies that accommodates simplicity by combining several low-dimensional fuzzy inference systems. Driven by two derivative-free optimization algorithms, a two-step tuning process is introduced to speed up the convergence process when traversing HFIT parameters. A Gaussian distributed resampling technique is developed to screen a small number of samples with diverse engine operations to maintain sample diversity. The experimental dataset is obtained from steady-state tests carried out in a BYD 1.5L gasoline engine specially made for a hybrid powertrain. The results demonstrate that the proposed fuzzy-tree-constructed data-efficient modelling methodology performs with superior learning efficiency on volumetric efficiency characterization than those of a fuzzy inference system, a neural network, or an adaptive neuro-fuzzy inference system. Even when dataset split ratio downs to 0.2, the relative mean absolute error can be restricted to 3.18% with the help of Gaussian distributed resampling technique.

Keywords

Data resampling; data-efficient modelling; dedicated hybrid engine; hierarchical fuzzy inference tree; volumetric efficiency.

1. Introduction

In the post-combustion engine era, as traditional automobile engines are accelerating innovation, many dedicated hybrid engines have appeared. Hybrid technology offers a bridge that connects transport electrification to flexible energy sources such as petrol [1] and ethanol [2]. Relying on the architecture of hybrid powertrains, they can continuously work with over 40% thermal efficiency to enhance engine torque, drivability and fuel-saving required by customers while meeting emission regulations [3]. The successes due to progress in advanced automotive electronics, such as variable valve actuation [4], exhaust gas recirculation [5], gasoline direct injection [6], variable geometry turbocharger [7]. These complex systems provide a large set of degrees of freedom available for engine regulation [8], but introduce many new control variables within the engine management system [9].

Air mass flow is one of the main control variables in the engine management system, which is determined by the volumetric efficiency. Pressure drops, gas temperature increases through heat transfer from the intake pipes and cylinder walls, gas inertia, overlapping valve and pressure waves at the intake manifold can disrupt cylinder filling [10–12]. Many authors have studied and modelled the intake air systems with the application of thermo-fluid dynamic governing equations. 1D wave action models are the most popular physical models in the intake air system analysis because of the tradeoff between accuracy and computational cost [13,14]. In fact, the process of thermo-fluid dynamics occurring in an internal combustion engine is so nonlinear and complex that it is unlikely to accurately characterize the whole process. Physically based modelling, although interpretable, requires specialized expertise, and its solving process is time consuming and unsuited for a control purpose [15]. In addition, many auxiliary electronic devices induce a significant increase in physical model complexity, resulting in difficulties for accurate implementation.

Rapid development in informatics has enabled fast modelling of complex physical systems based on the measurement of real-world performance [16]. Volumetric efficiency is commonly modelled empirically as a black-box function of a combination of engine speed, intake manifold pressure, intake manifold temperature, and exhaust manifold pressure [17]. Artificial neural networks (ANNs) have become an important tool in empirical engine process modelling [18]. Francesco et al. re-design the experimental campaign of volumetric efficiency model based on the use of NNs [19]. The research demonstrates that the calibration performance falls within acceptable limits even after a 60% cut of the experimental data usually acquired for calibration purposes. Luján et al. develop an adaptive learning algorithm to increase hidden layer weight update speed for NN-driven volumetric efficiency model [20], wherein this algorithm performs higher learning speed, reduced computational resources and lower network complexities. As a representative of heuristic approaches, the fuzzy inference system is widely used in engine modelling and control due to its excellent self-interpretability and robustness [21–23]. As a hybrid of the first two methods, the adaptive neuro-fuzzy inference system has the potential to capture the benefits of both in a single framework [24]. Since the added rule base extends neuron numbers in the hidden layer, the applications on modelling of emissions and heat transfer shows considerable prediction accuracy [25–27]. However, the input variables of these fuzzy-based model are quite limited (≤ 4) due to the computational burden of exponentially increasing rules. Such models with poor generalization are extremely difficult to apply in practice. In the case of involving multiple input variables (> 5), there is no discussion on the volumetric efficiency model performance.

Reducing the operation cost of experimental data is another vital challenge for data-driven modelling approaches. Zhou et al. research a transferable representation modelling routine to reduce the development workload for energy management controllers, where two artificial intelligence technologies of deep neural network [28] and Gaussian process regression [29] are developed to cooperate with an adaptive neuro-fuzzy inference system for knowledge transfer of the energy management controller. For data-driven engine modelling, most data collection procedures are based on blind scanning or uniform scanning in a very narrow window of limited variables [20]. That may cause its conclusion to be inconsistent with real working scenarios. There is an alternative way for the identification of the engine volumetric efficiency map using transient condition data [30], but as an increase of input variables, its look-up table would have a curse of dimensionality. Li et al. propose a novel approach of geometric neuro-fuzzy transfer learning to model in-cylinder pressure of a diesel engine that only utilizes limited experimental data

obtained by geometric screening to learn a high-precise transfer model [31]. For training complex models with wider ranges and more variables to adapt to real working scenarios, however, these procedures may cause the amount of data required to increase exponentially [16]. In fact, most of the collected experimental data has no value because they are not the key knee points for determining the functional relationship between input and output [32], and as concluded in [19]. Effectively distinguishing and cleaning up unnecessary training data for real working scenarios is worth the investment.

To minimize experimental effort on sample collection and maintain high-precision volumetric efficiency characterization, this paper proposes the new methodology of fuzzy-tree-constructed data-efficient modelling to precisely quantify the air mass flow through the engine. The experimental dataset is obtained from steady-state tests carried out in BYD 1.5L gasoline engines specially made for a hybrid powertrain. The three main contributions drawn from the investigation are:

1) Three topologies of HFITs are originally designed that accommodates simplicity by combining several low-dimensional fuzzy inference systems.
2) A two-step tuning process driven by two derivative-free optimization algorithms is introduced to speed up the convergence process while avoiding falling into a local minima when traversing HFIT parameters.
3) A Gaussian distributed resampling technique is developed to screen a small number of samples with diverse engine operations in order to maintain sample diversity and avoid its impoverishment in the volumetric efficiency model training.

The paper outline is as follows. Section 2 details the experimental laboratory set up, showing the features of the measurement equipment. Section 3 introduces technical background a fuzzy inference system that will be used as a basic unit for the proposed methodology development. In Section 4, the proposed solution of fuzzy-tree-constructed data-efficient modelling methodology is described, which comprises fuzzy tree construction, rule leaning and parameter tuning, and Gaussian distributed resampling. Section 5 carries out a comparative study of the proposed methodology and other mainstream machine learning models. Finally, conclusions are drawn in Section 6.

2. Experimental Setup

The experiments were conducted with an in line 4-cylinder, 1.5 L gasoline engine. The features of the engine are shown in Table 1. The engine was run under steady state conditions at different operating points that covered the entire engine torque and speed range. The experimental setup is in three main parts: the test cell, the experimental data, and error analysis.

Table 1. Engine specifications

Parameter	Value	Unit
Cylinder number	4	-
Bore × stroke	72*92	mm
Displacement	1498	cm^3
Compression ratio	15.5	-
Injection system	PFI	-
Maximum power	81/6000	$\mathrm{kW/rpm}$
Maximum torque	135/4500	$\mathrm{Nm/rpm}$
Torque at maximum power	129	Nm

2.1. Test Cell Description

The engine layout is shown in Fig. 1, where the engine operating variables used for the implementation of the proposed methodology i.e., fuzzy-tree-constructed data-efficient modelling are marked with blue arrows at the point of measurement. Relevant variables needed for volumetric model implementation were recorded, such as: engine speed, torque, intake manifold pressure, turbine inlet pressure, intake manifold temperature, EGR position and air mass flow rate. To assess the volumetric efficiency according to real engine conditions, EGR was performed depending on the engine load of the different running points.

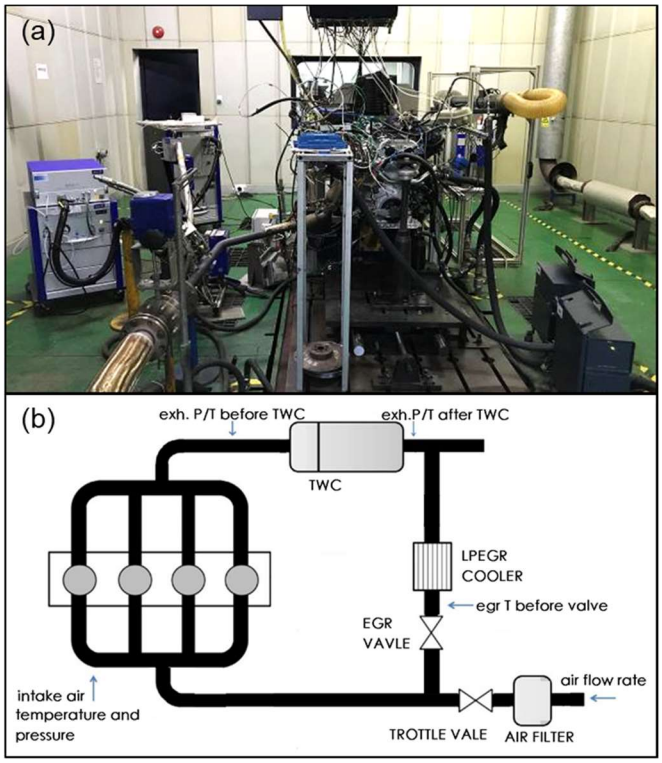

Fig. 1. Dedicated hybrid engine: a) testing bench and b) principal diagram

2.2. Experimental Data

Considering the diversity of experiment dataset (e.g., system noise and operators), engine experiment is carried out on the two testing benches for the same type dedicated hybrid engine [3] developed by BYD Auto Ltd. 1889 samples were collected at different steady engine load conditions of the operating range of the studied engine. The range of the operating points was as follows: 1000-6000 rpm for engine speed, 1.5 -135 N m for engine torque, and 0–100% for LP EGR positions. Engine speed variables at each steady state point were obtained from the average of 600 points sampled at 10 Hz. Measured volumetric efficiency over various torque regions is shown in Fig. 2.

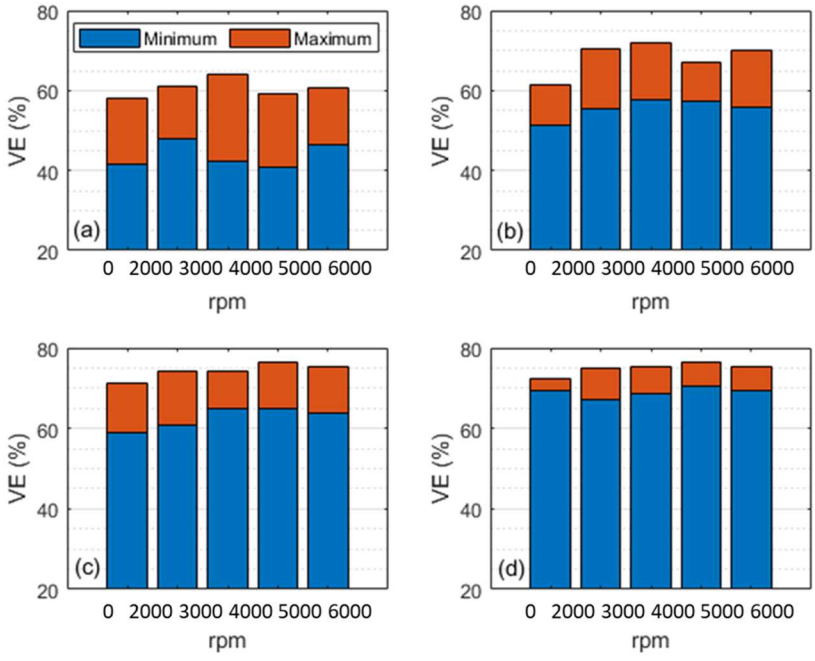

Fig. 2. Measured volumetric efficiency over various torque regions: a) 0-50 Nm; b) 50-75 Nm; c) 75-100 Nm; and d) 100-135 Nm

2.3. Error Analysis

Errors in experimental studies and the resultant measurements occur and include both random and systematic errors. The sources of these errors are wide ranging and include experimental planning, instrument selection, condition of the instrument, calibration, environmental conditions, observation of measurement, and measurement reading. Therefore, to demonstrate the accuracy of experimental studies, uncertainty analysis can be used. The random error refers to an accidental and unanticipated change in the experimental conditions, for example, mechanical variation, electrical interference, sensor or temperature variation. The systematic error is non-random and is determined as the difference between the actual value and the mean value. In this project, the experimental error and uncertainty were calculated by using the statistical tolerance analysis method Root Sum of Squares (RSS), using the following equation:

$$RSS = \sqrt{(\varepsilon s)^2 + (2\varepsilon r)^2} \tag{1}$$

where εs is a system error, and εr is a random error. The relative percentage uncertainties of various parameters are calculated, as shown in Table 2.

Table 2. Signal measurement and testing facilities

Signal parameters	Uncertainties	Signal parameters	Uncertainties
Speed	± 1 rpm	Intake pressure	± 0.25 %
Torque	0.1 Nm	Fuel consumption	± 1 %
In-cylinder pressure	0.2 %	Fuel pressure	± 0.25 %
Exhaust temp.	± 15 ℃	EGR	± 2 ℃
Intake temp.	± 2 ℃	Exhaust pressure	± 0.25 %

3. Fuzzy Inference System

Fuzzy inference systems (FISs) have been developed by Zadeh to handle uncertainty and imprecision in decision making process for real world applications. For engine performance prediction, this system has been proved that has simpler system design, development, lower cost, and greater ease of maintenance [33]. The standard FIS is composed of the following components [34]: 1) a fuzzifier that fuzzifies the input data; 2) a knowledge base, which contains fuzzy rules in the form of IF-THEN terms 3) an inference engine, which calculates firing strengths of the rules to infer knowledge from the knowledge base; and 4) a defuzzifier that converts the inferred knowledge into rule actions. The knowledge base of FIS consists of a database and a rule base. The database assigns fuzzy sets to input variables, and the fuzzy sets convert the input variables into membership function degrees. For rule induction, the rule base constructs a set of rules to obtain fuzzy sets from the database.

As compared to Mamdani systems, this research uses Takagi–Sugeno–Kang (TSK)-type FIS objects for faster evaluation during the tuning process of hierarchical fuzzy inference trees (HFITs), which governed by IF–THEN rules of the form [35]

$$R_i\text{: IF } x_1 \text{ is } A_{i1} \text{ AND } \dots \text{AND } x_{d^i} \text{ is } A_{id^i} \text{ THEN } y \text{ is } B_i \tag{2}$$

where R_i is the ith rule in an FIS, $A_{i1}, \dots, A_{id}$ are the fuzzy sets, B_i is a function of the input vector $x = \langle x_1, x_2, \dots, x_{d^i} \rangle$ that returns a crisp output y, and d^i is the total number of the inputs presented to the ith rule. Note that the number of inputs may vary from rule-to-rule. Therefore, the dimension of inputs in a rule is denoted as d^i. For TSK type, the function B_i is usually described as

$$B_i = c_i^0 + \sum_{j=1}^{d^i} c_i^j x_j \tag{3}$$

where c_j^i for $j = 0$ to d^i are the free parameters in the consequent part of a rule. The defuzzified crisp output of FIS is computed as follows: First, the inference engine fires up the fuzzy logic rules. The firing strength f_i of the ith rule is computed as

$$f_i = \prod_{j=1}^{d^i} \mu A_{ij}(x_j) \tag{4}$$

where μA_{ij} is the degrees of jth fuzzy set membership function at the ith rule. Then, the defuzzified output y of an FIS is computed as

$$y = \frac{\sum_{i=1}^{M} B_i f_i}{\sum_{i=1}^{M} f_i} \tag{5}$$

where M is the total rules in the rule base. In this paper, the fuzzy set A was of the form as considered in [36]

$$\mu A(x) = \frac{1}{1 + \left(\frac{x - m}{\sigma}\right)^2} \tag{6}$$

where m and σ are the centre and the width of membership function $\mu A(x)$, respectively.

4. Fuzzy-Tree-Constructed Data-efficient Modelling Methodology

In order to minimize experimental effort on sample collection and maintain high-precision volumetric efficiency characterization, the new methodology of fuzzy-tree-constructed data-efficient modelling is proposed which includes three main procedures: 1) tree construction, where three fuzzy tree topologies are designed based on depth of fuzzy trees; 2) rule learning and parameter tuning, where the rule base is learnt while keeping the scalar parameters of membership functions constant first and then the scalar parameters of membership functions and rules are tuned. Expectedly, the proposed methodology should be sufficiently general to guide the multivariate modelling of generic engines.

4.1. Fuzzy Tree Construction

An HFIT is an optimum tree-based system that accommodates simplicity by combining several low-dimensional FISs [36]. Its hierarchical structure is analogous to a multilayer feedforward NN, where the nodes (the low-dimensional FISs) are connected using weighted links. The concept of forming an HFIT is inherited from the flexible neural tree proposed by Chen et al. [37]. Usually, constructing a FIS tree uses the following steps: 1) rank the input attributes based on their correlations with the output attribute; 2) create multiple FIS objects using the ranked input attributes; and 3) construct a FIS tree from the FIS objects. Fig. 4 shows the statistical results of correlation analysis for this seven-input-one-output experimental dataset. In the final row of the correlation matrix, the first seven elements show the correlation coefficients between the seven input data attributes and the output attribute.

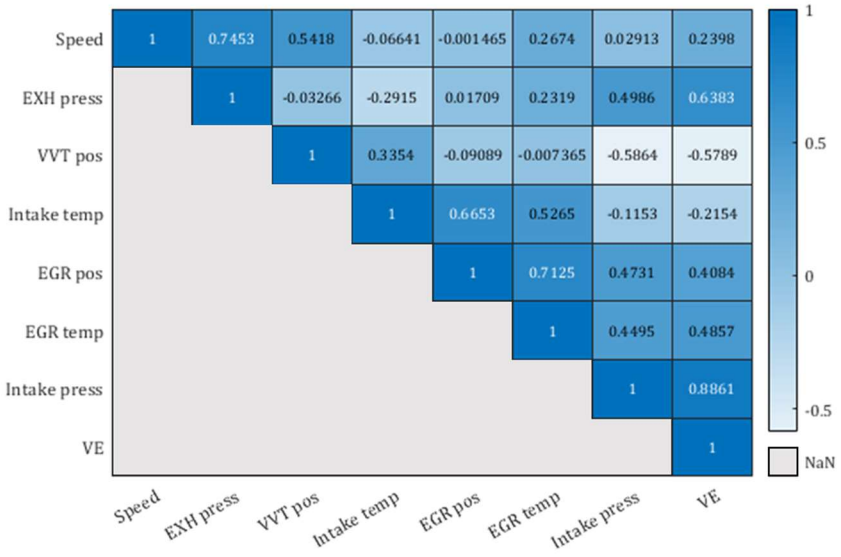

Fig. 3. Statistical results of correlation coefficients

Because the HFIT consists of multiple two-input-one-output FIS objects, input attributes with negative and positive correlation values are paired up to combine both positive and negative effects on the output for volumetric efficiency prediction. Thus, the inputs are sorted according to their ranks: intake pressure, x_1, VVT position, x_2, exhaust pressure, x_3, intake temperature, x_4, EGR temperature, x_5, speed, x_6, EGR position, x_7.

In order to reduce the complexity of the rule base, three tree topologies are originally designed for volumetric efficiency modelling as shown in Fig. 4, namely incremental, cascaded, and aggregated trees. In an incremental structure, input values are incorporated in multiple stages to refine the output values in several levels. A cascaded structure combines both incremental and aggregated structures to construct a fuzzy tree. In an aggregated structure, input values are incorporated as groups at the lowest level, where each input group is fed into a FIS.

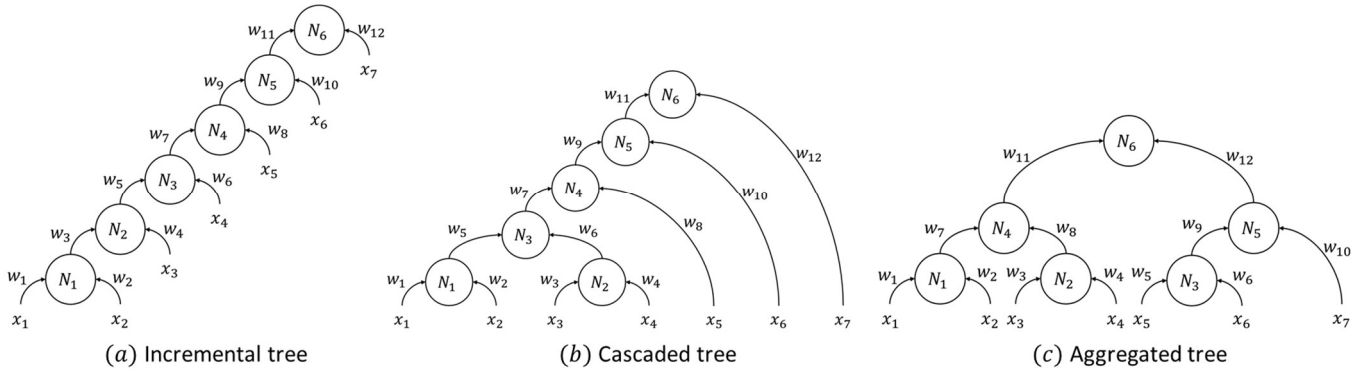

Fig. 4. Three designed topologies of HFITs: a) incremental; b) cascaded; and c) aggregated trees

Each node N_i in an HFIT receives a weighted input $x_i w_i$, where w_i is the weight. In this paper, however, the weights in HFIT were set to 1 because the objective of this paper was also to reduce the complexity of the produced tree along with the approximation error. Setting weights to 1 also allows raw input to be fed to the fuzzy sets. Besides, each input contains two standard triangular membership functions so that the output includes four membership functions. Because each HFIT topology has six nodes i.e., FISs, the total rules to be tuned for each HFIT topology is 24.

4.2. Rule Learning and Parameter Tuning

In order to improve the learning efficiency of HFITs, we introduce the two-step tuning process as illustrated in Fig. 5, 1) learning the rule base while keeping the input and output MF parameters constant; and 2) tuning the parameters of the input/output MFs and rules. The first step is less computationally expensive due to the small number of rule parameters, and it quickly converges to a fuzzy rule base during training. In the second step, using the rule base from the first step as an initial condition provides fast convergence of the parameter tuning process. Since the FIS tree already learned rules from the training data, using a local optimization method will yield fast convergence of the parameter values. Tuning the FIS tree parameters takes more iterations than the previous rule-learning step.

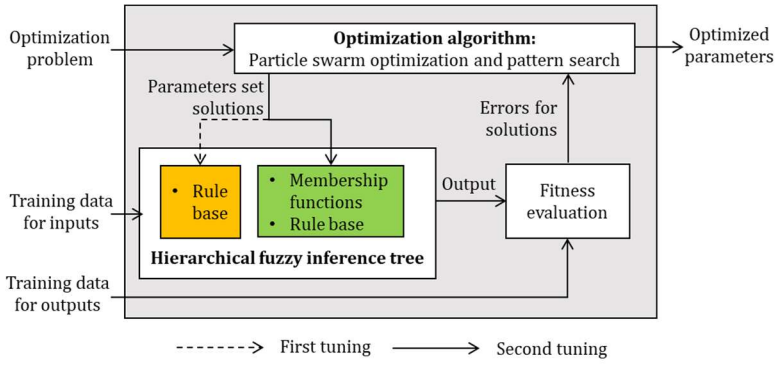

Fig. 5. Workflow of the two-step tuning process for HFITs

During the tuning process, the maximum number of rules for each FIS is restricted to 4. The number of tuned rules of each FIS can be less than this limit, since the tuning process removes duplicate rules. To avoid falling into a local minima when traversing HFIT parameters, two derivative-free optimization algorithms of particle swarm optimization (PSO) and pattern search (PS) have been investigated in this two-step tuning process. Their search mechanisms are illustrated in Fig.6.

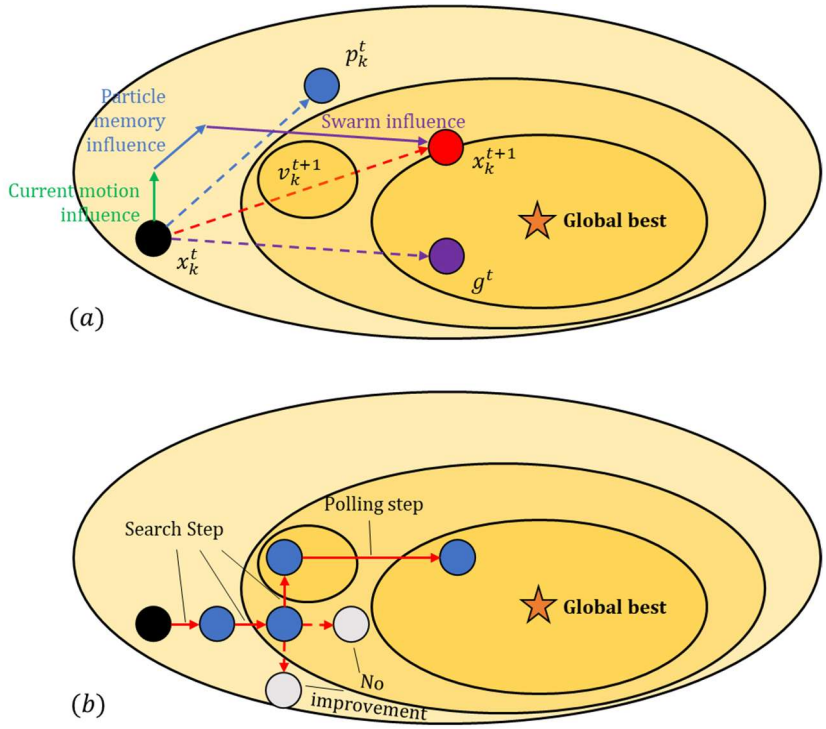

Fig. 6. Search mechanisms of: a) PSO and (b) PS algorithms

The PSO is a global optimization algorithm that uses particles to discover the search space and converge on the global minimum [38], where the algorithm spreads these particles over the search space and iteratively moves their location based on previous best locations and stochastic variables. Considering a single swarm of size K, connected completely, and an *N*-dimensional search space, the velocity, and position of each particle are updated as illustrated in Fig. 4(a) and follows:

$$v_{k,n}^{t+1} = wv_{k,n}^t + c_1 r_1 \left(p_{k,n}^t - x_{k,n}^t\right) + c_2 r_2 \left(g_n^t - x_{k,n}^t\right) \tag{7}$$

$$x_{k,n}^{t+1} = x_{k,n}^t + v_{k,n}^{t+1} \tag{8}$$

where $1 \le k \le K$ and $1 \le j \le N$; x_k^t and v_k^t are the position and velocity vector of the kth particle at the tth time step, respectively; p_k^t is the personal best position of the kth particle at the tth time step, whereas g^t is the global best position of all particles in the swarm at the tth time step; r_1 and r_2 are random numbers uniformly distributed in the range $[0,1]$; w is the inertia weight, whereas c_1 and c_2 are the cognitive coefficients. The weight with inertia is a key factor for the PSO convergence process that works to keep the balance between the exploration and exploitation of the exploration space. For simplifying the implementation of PSO, the weight with inertia may be fixed to a constant in this case.

Different from the PSO, the PS is a local optimization algorithm that requires fewer computing resources but evaluates points only within a certain proximity to the current location [39], where the algorithm uses a pattern of search directions and searches around the current point using the directions of the pattern and the size of the current mesh size. In conjunction of Fig. 4(b), the pseudocode of the PS algorithm can be described as:

1 Initialize the mesh, matrix G for the mesh, and the matrix H containing additional search directions, the step size, starting location, and the termination conditions.
2 While (termination conditions not met):

Search: Test function values at new locations by using a subset of positive spanning matrix. If a better value in these locations yields detected, set $x(t+1)$ equal to this location, optionally increase the step size, and repeat this step **Search**. Otherwise, go to step **Poll**.
Poll: test function values around the current location by using the matrix G. If a better value in these locations yields detected, set $x(t+1)$ equal to this location, optionally increase the step size, and return to step **Search**. Otherwise, decrease step size, increase failed poll step counter by 1, and return to step **Search**.

Usually, a learning algorithm has a single objective (approximation error minimization) that is often achieved by minimizing the root-mean-squared error (RMSE) on the learning data.

$$E = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_i - y_i)^2} \tag{9}$$

where d and y are the desired and the model's outputs, respectively, and N is the number of data pairs in the training set. In this paper, the selection of the best phenotype in a single objective training was solely based on a comparison of the RMSEs.

4.3. Gaussian Distributed Resampling

To minimize experimental effort on large database collection, a Gaussian distributed resampling technique is developed to screen a small number of representative samples with a wider engine operation range for maintaining high-precision volumetric efficiency characterization. The key concept of the proposed technique is to generate re-sampling locations on the basis of a Gaussian distribution [40], which depends on the size of the weights in the resampling process. In comparison with the industry-standardized experimental design process, sample diversity can be maintained, and thus the proposed technique avoids sample impoverishment. Predictively, it can guarantee a reliable estimation even if the number of samples is sharply reduced. The workflow of Gaussian distributed resampling is illustrated in Fig. 7 and its detail is as follows.

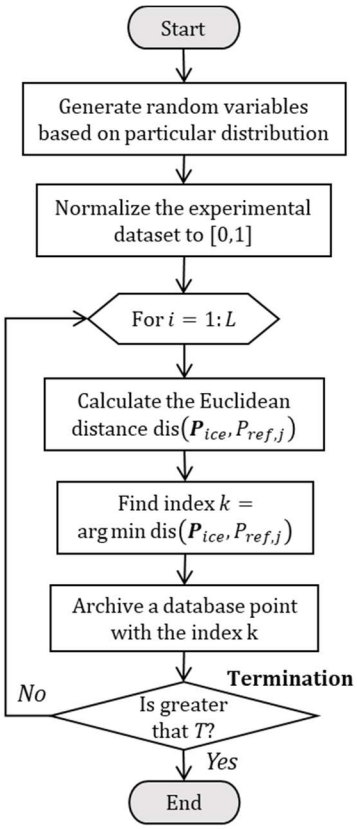

Fig. 7. Workflow of Gaussian distributed resampling technique

To facilitate the implementation of Gaussian distributed resampling technique, as the most used control variables in the engine map calibration process, speed and torque are selected as resampling control variables, and their experimental dataset both needs to be normalized to [0,1].

$$\left.\begin{aligned} n'_{ice,i} &= \frac{n_{ice,i} - \min(\boldsymbol{n}_{ice})}{\max(\boldsymbol{n}_{ice}) - \min(\boldsymbol{n}_{ice})} \\ T'_{ice,i} &= \frac{T_{ice,i} - \min(\boldsymbol{T}_{ice})}{\max(\boldsymbol{T}_{ice}) - \min(\boldsymbol{T}_{ice})} \end{aligned}\right\} \tag{10}$$

where, $n_{ice,i}$ and $n'_{ice,i}$ are ith speeds before and after normalization; $T_{ice,i}$ and $T'_{ice,i}$ and ith are torques before and after normalization; $\boldsymbol{n}_{ice}$ and $\boldsymbol{T}_{ice}$ are the vectors of speed and torque collected from the engine experiment.

Four sampling patterns are studied and used to generate random variables as sampling reference points. In terms of the mean and standard deviation of Gaussian distribution, $\mu = [0,0.5,1]$ is chosen for mean that represents low-power, medium-power, high-power zones of the studied engine, respectively. Choice of standard deviation is subject to original experimental sample distribution. After investigation on $\sigma^2 = [0.2,0.5,1,2,3]$ for standard deviation, standard deviation is fixed to 1 because it performs better the fitting performance when conducting the Gaussian distributed resampling technique. Besides, the widely-used continuous uniform distribution ($X \sim U[a,b]$, for $a = 0$ and $b = 1$) also has been considered in comparison. Because each new sampling reference point cannot find a completely coincident point in the experimental database, Euclidean distance defined as Eq. (11) is invoked to search for points in the experimental database that have the shortest distance. For jth sampling reference point $P_{ref,j}$, Euclidean distances can be expressed as

$$\mathrm{dis}\left(\boldsymbol{P}_{ice}, P_{ref,j}\right) = \sqrt{\left(\boldsymbol{n}_{ice} - n_{ref,j}\right)^2 + \left(\boldsymbol{T}_{ice} - T_{ref,j}\right)^2} \tag{11}$$

where, $n_{ref,j}$ and $T_{ref,j}$ are speed and torque of jth sampling reference point. The index of the database point with minimum Euclidean distance is computed as

$$k = \arg\min \mathrm{dis}\left(\boldsymbol{P}_{ice}, P_{ref,j}\right) \tag{12}$$

Once the point with minimum Euclidean distance is detected, its information will be retrieved via the pointer, and then recorded. This process uses alternative sampling i.e., the recorded point will be removed from the original database to avoid entry duplication. Finally, the data clean process will end when iterations i meet the termination condition L.

5. Results and Discussion

From here on, a comprehensive comparative study is carried out from the three aspects of 1) fuzzy tree topology; 2) effect of dataset split ratio; and 3) performance of Gaussian distributed resampling.

5.1. Fuzzy Trees with Different Topologies

This section investigates the impact of fuzzy tree topologies on the prediction performance and effectiveness of learning and tuning algorithms. Before comparing the fuzzy tree topologies, suitable learning and tuning algorithms must be determined, and they should have computational efficiency and an ability to find global optima in industrial practice. Thus, three algorithm combinations of learning and tuning are contrasted, which are composed of global and local optimization algorithm representatives i.e., PSO and PS. For each algorithm combination, dataset split ratio of training and testing sets to 1:1 and repetitive training is performed 20 times.

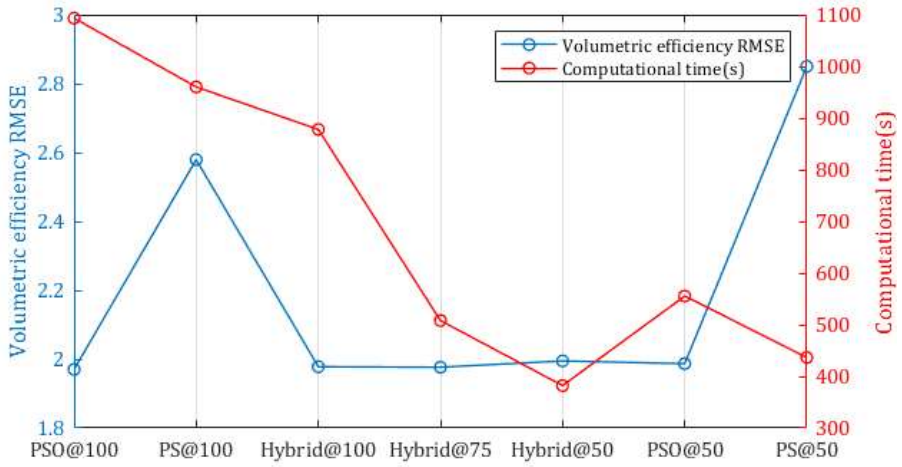

Fig. 8. volumetric efficiency prediction performance and training time comparison of using different algorithm combinations

Fig. 8 organizes volumetric efficiency RMSE and training time under different algorithm combinations of learning and tuning, in which PSO@100 denotes that PSO is applied for both learning and tuning processes and iteration terminal is 100 for each. Similar to Hybrid@50, it denotes that PSO and PS are applied for learning and tuning processes separately and iteration termination is 50 for each. Compared with PSO@100 and PSO@50, the hybrid combinations with 50-100 iterations performed competitively and stably, and do not reduce the accuracy of the model due to the addition of a local optimization algorithm. However, the training time of the PSO combination has increased in varying degrees, reaching a maximum of 45.2% (PSO@50). Although PS combinations have relatively low training time, volumetric efficiency RMSE is not satisfactory and at least 30.3% growth compared to Hybrid@100. Considering the computational efficiency factor, this paper adopts Hybrid@75 for hierarchical fuzzy inference trees in later discussion.

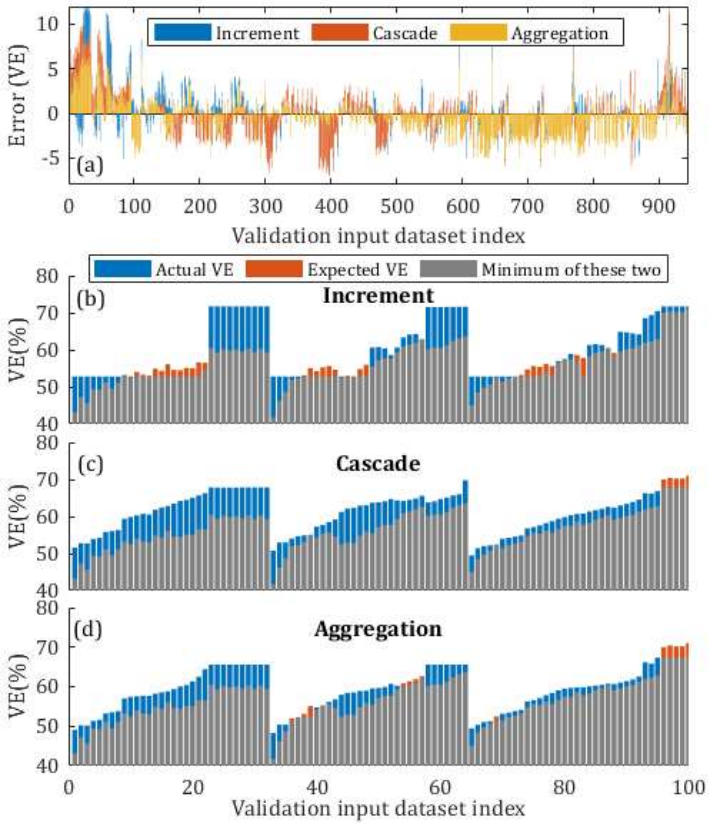

Fig. 9. Prediction performance comparison of hierarchical fuzzy inference trees: a) volumetric efficiency error of validation dataset; and performance detail at index segment [0,100] of b) incremental tree; c) cascaded tree; and d) aggregated tree

Fig. 9 shows the prediction performance comparison of hierarchical fuzzy inference trees with different topologies. In Fig. 9(a), the maximum error of the aggregated fuzzy tree is significantly lower than those of the incremental and cascaded trees, especially at the index segment [0,100]. As shown in Fig. 9(b), the incremental tree has a step effect in the function approximation of the index segment. Fig. 9(c)(d) illustrates that the incremental and aggregated trees are relatively smooth and closer to the trend of the experimental results. The authors believe that this is due to the tuning strides of the level parameters not all being on the same scale because each input is added to different nodes in the incremental tree. Consequently, some high-level parameters with overweight may result in homogenized input value characteristics. Table 3 summarizes the numerical evaluation results of three hierarchical fuzzy inference trees. It is worth mentioning that the maximum error of using the aggregated fuzzy tree is much lower than that of using incremental and cascaded trees, reducing it by up to 51.3%.

Table 3. Fuzzy tree prediction performance comparison of using PSO and PS combination

Tree topology	Maximum error	RMSE	R^2
Increment	12.77	1.9843	0.933
Cascade	11.79	2.1297	0.898
Aggregation	6.22	1.7652	0.968

5.2. Effect of Dataset Split Ratio on Machine Learning Models

To reduce experimental effort on sample collection, this section investigates the prediction behaviours of using three typical machine learning models i.e., FIS, NN, and ANFIS. Various dataset split ratios of training and testing from 0.1 to 0.8 are considered to test the robustness of the studied machine learning models, especially with the small sample size. The repeatability experiment is also carried out 20 times and all samples are selected based on the continuous uniform distribution.

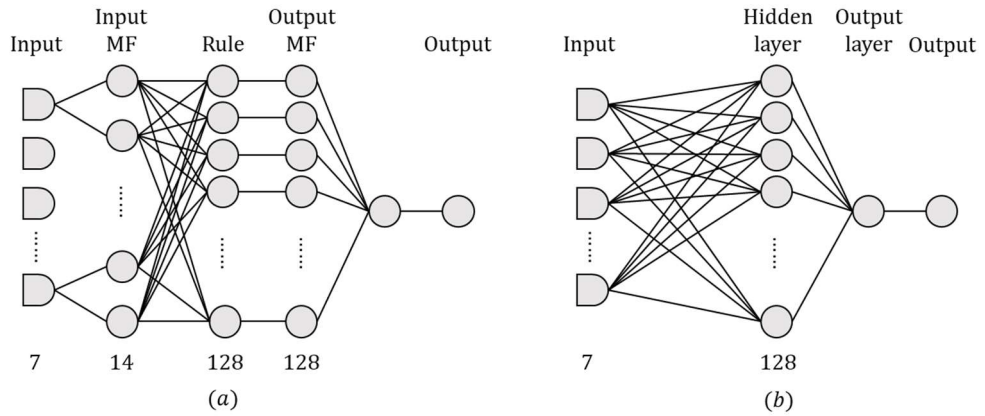

Fig. 10. Configuration of the studied machine learning models: a) ANFIS model and b) NN model

In order to fairly evaluate prediction performance of each machine learning models, their model complexity should be appropriately restricted in the same level. The FIS has the same basic structure and parameter tuning procedures as the proposed HFIT's, the main difference is that the FIS is the only model structure to characterize the relationship of all inputs and one output. On the basis of the FIS structure, the ANFIS and NN are developed with the full connected network architectures (as shown in Fig. 10(a) and (b)) that introduces hidden layers to enhance nonlinear regression performance.

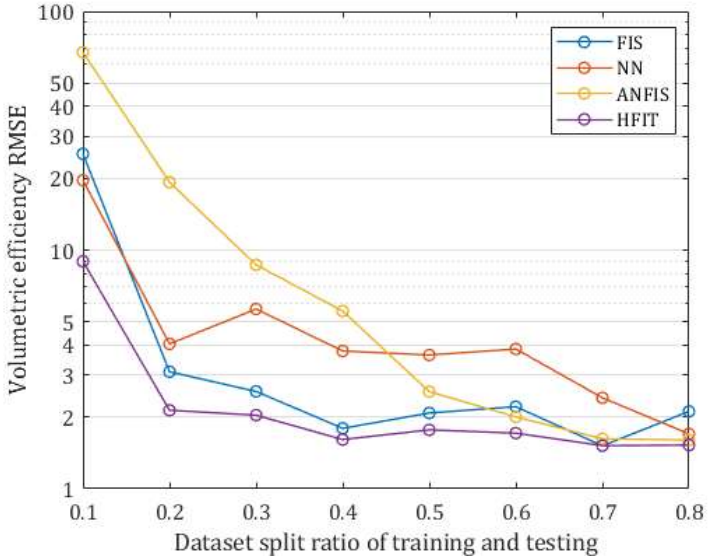

Fig. 11. Volumetric efficiency prediction performance comparison under various dataset split ratios

Fig. 11 shows a volumetric efficiency prediction performance comparison of FIS, NN, ANFIS, and HFIT under various dataset split ratios. When applying a high dataset split ratio ($\lambda = 0.8$), NN, ANFIS, and proposed HFIT perform good performance of volumetric efficiency function approximation. In general, the RMSE of volumetric efficiency prediction has gradually increased as the dataset split ratio decreases. It is worth mentioning that two fuzzy-based models still show strong robustness on volumetric efficiency function approximation, compared to NN and ANFIS models, even when the dataset split ratio falls to 0.2. However, RMSE of volumetric efficiency prediction by using ANFIS is growing exponentially (RMSE=67.374 when $\lambda = 0.1$) with dataset split ratio goes down. The numerical results for the four machine learning models are shown in Table 4.

Table 4. RMSE comparison of using four machine learning models under various dataset split ratios

ML model	RMSE under various dataset split ratios							
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
FIS	25.320	3.0915	2.5577	1.7917	2.0777	2.2082	1.5139	2.1095
NN	19.622	4.0465	5.6647	3.7789	3.6341	3.8514	2.4050	1.6989
ANFIS	67.375	19.2532	8.6726	5.5554	2.5523	2.0012	1.6203	1.6001
HFIT	8.9740	2.1331	2.0341	1.6080	1.7652	1.7107	1.5162	1.5240

The distribution of predicted and measured volumetric efficiency widens considerably when the dataset split ratio is below 0.2. Fig. 12 shows machine learning model outcomes of the training data plotted against their respective target values. Dotted lines represent a 5% relative error above and below the measured value. Higher dispersion is observed in the NN and ANFIS cases, which show a higher number of points that are outside the boundary of the 5% relative error. In the case of using FIS, volumetric efficiency values below 58% are not learnt so the predicted value does not change when the input changes below 58%. In the case of using the proposed HFIT, a higher number of the predicted volumetric efficiency are located inside or closer to the boundary of the 5% relative error. Therefore, the proposed HFIT has great potential in reduced sample modelling. On this basis, the authors intend to further explore the effect of experimental data reduction in different zones or densities on the accuracy of the model.

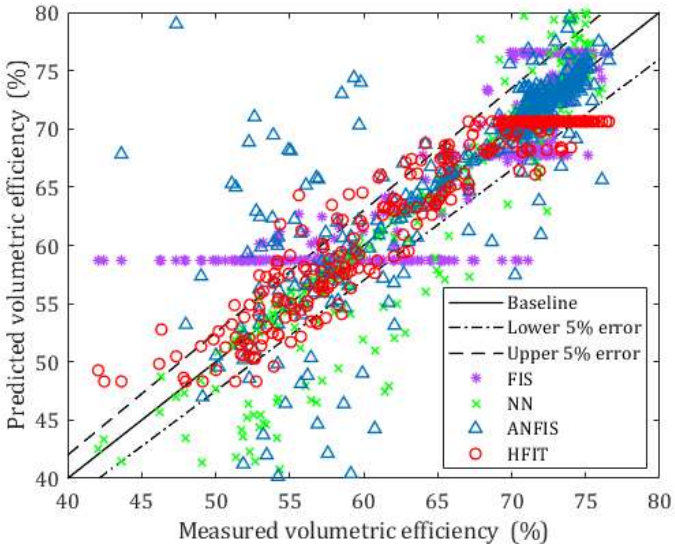

Fig. 12. Predicted and measured volumetric efficiency of using four studied machine learning models with dataset split ratio $\lambda = 0.2$

5.3. Performance Assessment of Gaussian Distributed Resampling

This section discusses how the Gaussian distributed resampling technique reduces experimental samples while maintaining the volumetric efficiency model accuracy. Fig. 13 illustrates the dataset distribution and mapping result achieved by applying Gaussian distributed resampling technique.

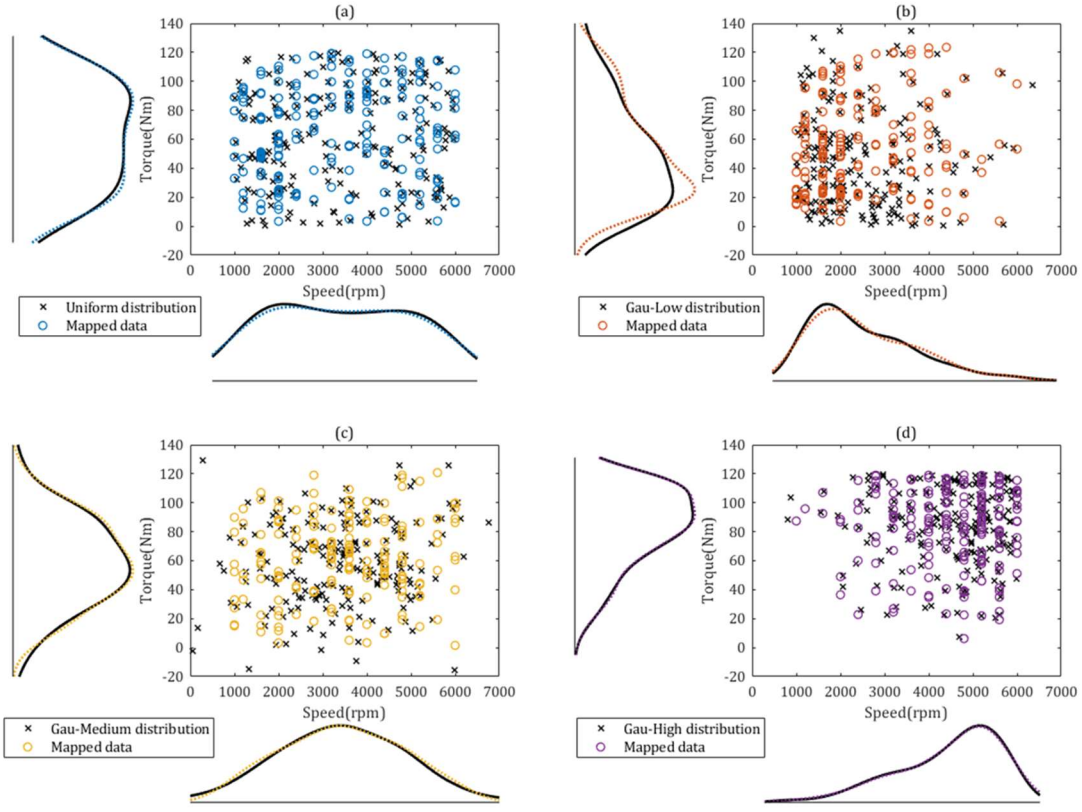

Fig. 13. Dataset distribution and mapping result after re-sampling

The four studied distributions (i.e., Uniform, Gau-Low, Gau-Medium, and Gau-High) are almost in line with the distributions of their mapped experimental datasets. In addition to the uniform distribution of Fig. 13(a), the distribution data of Fig. 13(b)-(d) are respectively concentrated in the low, medium, and high-power zones. Some native random variables are outside the operating range of the dedicated hybrid engine. This technique can effectively constrain them within their operating ranges and maintains the original distribution curve through the Euclidean distance determination. The use of other control variables may

reduce the experiment cost, but its applicability is limited by too many factors, such as: parameter controllability and working ranges of various engines.

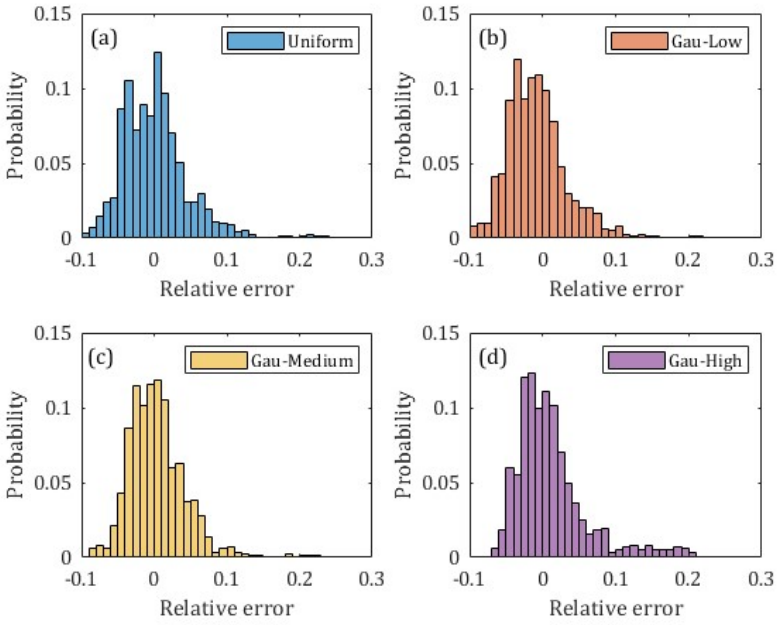

Fig. 14. Validation relative error distributions of using Gaussian distributed resampling based on a) Uniform; b) Gau-Low; c) Gau-Medium and (d) Gau-High distributions

From Fig. 14, the distribution intervals of Uniform, Gau-Low, and Gau-Medium can be regulated in $[-0.1,0.1]$. however, the positive distribution interval of Gau-High extends to 0.2. It is difficult to characterize volumetric efficiency through the experimental database of concentrated sampling in the high-power zone. This may be because the non-linear relationship between the parameters in the low and medium power zones is higher and its adjustment is more sensitive. In conjunction with Table 5, the developed Gaussian distributed resampling technique has concentrated sampling in the medium-power zone, and has the best overall performance of the four evaluation metrics including coefficient of determination (R^2), RMSE, mean absolute error (MAE), and relative MAE (RMAE). Compared with the number of boundary samples (low or high-power zones), the number of centre samples has a greater impact on volumetric efficiency characterization. In general, the developed Gaussian distributed resampling technique can effectively clean up 80% of the experimental data while maintaining the RMAE of 3% in the HFIT-assisted volumetric efficiency modelling.

Table 5. Prediction performance of using Gaussian distributed resampling

Evaluation metric	Uniform	Gaussian-Low	Gaussian-Medium	Gaussian-High
R^2	0.886	0.890	0.897	0.871
RMSE	2.7094	2.6567	2.6420	2.9578
MAE	2.1226	2.1037	2.0410	2.1196
RMAE	3.31%	3.29%	3.18%	3.29%

In order to further explore the real performance of the proposed fuzzy-tree-constructed data-efficient modelling methodology (with 20% of experimental data selected by Gaussian distributed resampling), Table 6 comprehensively assesses volumetric efficiency prediction performance of the studied dedicated engine in each engine operation regions. The areas with low error (RMAE>1%) are concentrated in the centre of engine operations, which would be attributed to the Gaussian distributed resampling technique (with Gau-Medium distribution). From a view of engine torque, volumetric efficiency prediction error (RMAE) in the low torque area (0-50 Nm) is larger than the error in the higher torque areas (>50 Nm). In conjunction with Fig. 4, authors believe the reason is that volumetric efficiency is greatly affected by the speed in the low torque area which increases regression difficulty of the proposed methodology.

Table 6. Volumetric efficiency prediction error (RMAE) comparison over various engine operation regions

Speed (rpm)	Torque (Nm)				
	0-50	50-75	75-100	100-135	Overall
0-2000	9.14%	1.53%	5.21%	0.86%	4.57%
2000-3000	4.25%	0.86%	0.80%	0.73%	1.21%
3000-4000	10.92%	1.47%	0.99%	0.75%	2.57%
4000-5000	15.64%	2.44%	0.86%	1.41%	3.73%
5000-6000	8.41%	4.23%	3.69%	1.83%	4.41%
Overall	9.59%	1.92%	2.18%	1.09%	3.18%

In terms of the generalization ability of the proposed methodology, it has great potential to be applied to other engine parameters, different engine configurations, and biofuels with various physicochemical characteristics e.g., brake thermal efficiency and emissions. The main challenge for implementing this methodology for various cases is to find the most effective features for such data-driven prediction systems. This would be the focus of our recent research.

6. Conclusions

This paper studies data-driven modelling for dedicated hybrid engine volumetric efficiency, wherein a new methodology of fuzzy-tree-constructed data-efficient modelling is proposed to precisely quantify the air mass flow through the engine. An extensive parametric study of the proposed methodology is carried out in terms of 1) fuzzy trees with different topologies; 2) effect of dataset split ratio on machine learning models; and 3) performance assessment on Gaussian distributed resampling. The conclusions drawn from the investigation are as follows:

1. The HFIT using aggregated topology has a minimum error of volumetric efficiency prediction that is much lower than that when using incremental and cascaded trees, reducing error by up to 51.3%.
2. In the process of HFIT optimization, the hybrid combination of PSO and PS has a lower RMSE (up to 45.2%) and training time (up to 30.3%) when compared to using just one of them.
3. For reduced dataset split ratios of training and testing, the proposed HFIT shows strong robustness on volumetric efficiency function approximation, compared to FIS, NN and ANFIS machine learning models, even when dataset split ratio is reduced to 0.2.
4. The developed Gaussian distributed resampling technique effectively cleans up 80% of the experimental data while maintaining the RMAE of 3% in the HFIT-assisted volumetric efficiency modelling.
5. To better implement Gaussian distributed resampling technique, increasing the number of centre samples has a greater impact on volumetric efficiency characterization than increasing the number of boundary samples in both low and high power zones.

The work presented in this paper is a significant step towards ‘data-driven data-efficient modelling methodology’. However, it is only a first step and neglects some aspects of a complete solution, and these will be the subject of future work. In terms of the generalization ability of the proposed approach, it shows great potential to be applied to other engine parameters, different engine configurations, and biofuels with various physicochemical characteristics. Besides, reducing experimental cost of map calibration by using transient data is an alternative way to collaboratively improve the experimental efficiency of this methodology. These all are worthy to be studied in the future work.

References

[1] Li J, Zhou Q, He Y, Williams H, Xu H. Driver-identified Supervisory Control System of Hybrid Electric Vehicles based on Spectrum-guided Fuzzy Feature Extraction. IEEE Trans Fuzzy Syst 2020;6706:1–1. https://doi.org/10.1109/tfuzz.2020.2972843.

[2] He Y, Wang C, Zhou Q, Li J, Makridis M, Williams H, et al. Multiobjective component sizing of a hybrid ethanol-electric vehicle propulsion system. Appl Energy 2020;266. https://doi.org/10.1016/j.apenergy.2020.114843.

[3] Lu G, Yang D, Rong Y, Gong Z, Wang B. Development of an Intelligent Thermal Management System for BYD DM-i Hybrid Engine. SAE Tech Pap 2021:1–10. https://doi.org/10.4271/2021-01-1153.

[4] Bozza F, De Bellis V, Teodosio L. A numerical procedure for the calibration of a turbocharged spark-ignition variable valve actuation engine at part load. Int J Engine Res 2017;18:810–23. https://doi.org/10.1177/1468087416674653.

[5] Yang C lei, Zu X huan, Wang HC, Wang Y yan. Optimized modelling and application of exhaust gas recirculation performance evaluation of turbocharged diesel engine. R Soc Open Sci 2018;5. https://doi.org/10.1098/rsos.172112.

[6] Frommater S, Neumann J, Hasse C. A phenomenological modelling framework for particle emission simulation in a direct-injection gasoline engine. Int J Engine Res 2020. https://doi.org/10.1177/1468087419895161.

[7] Bahiuddin I, Mazlan SA, Imaduddin F, Ubaidillah. A new control-oriented transient model of variable geometry turbocharger. Energy 2017;125:297–312. https://doi.org/10.1016/j.energy.2017.02.123.

[8] Li J, Gu Y, Wang C, Liu M, Zhou Q, Lu G, et al. Pedestrian-Aware Supervisory Control System Interactive Optimization of Connected Hybrid Electric Vehicles via Fuzzy Adaptive Cost Map and Bees Algorithm. IEEE Trans Transp Electrif 2021;7782:1–1. https://doi.org/10.1109/tte.2021.3124606.

[9] Shrivastava N, Khan ZM. Application of Soft Computing in the Field of Internal Combustion Engines: A Review. Arch Comput Methods Eng 2018;25:707–26. https://doi.org/10.1007/s11831-017-9212-9.

[10] Mezher H, Chalet D, Migaud J, Chesse P. Frequency based approach for simulating pressure waves at the inlet of internal combustion engines using a parameterized model. Appl Energy 2013;106:275–86. https://doi.org/10.1016/j.apenergy.2013.01.075.

[11] Feng H, Wang X, Zhang J. Study on the effects of intake conditions on the exergy destruction of low temperature combustion engine for a toluene reference fuel. Energy Convers Manag 2019;188:241–9. https://doi.org/10.1016/j.enconman.2019.02.090.

[12] Park C, Ebisu M, Bae C. Effects of Turbocharger Rotation Inertia on Instantaneous Turbine Efficiency in a Turbocharged-Gasoline Direct Injection (T-GDI) Engine. J Eng Gas Turbines Power 2021;143:1–11. https://doi.org/10.1115/1.4049299.

[13] Cornolti L, Onorati A, Cerri T, Montenegro G, Piscaglia F. 1D simulation of a turbocharged Diesel engine with comparison of short and long EGR route solutions. Appl Energy 2013;111:1–15. https://doi.org/10.1016/j.apenergy.2013.04.016.

[14] Duan X, Liu Y, Liu J, Lai MC, Jansons M, Guo G, et al. Experimental and numerical investigation of the effects of low-pressure, high-pressure and internal EGR configurations on the performance, combustion and emission characteristics in a hydrogen-enriched heavy-duty lean-burn natural gas SI engine. Energy Convers Manag 2019;195:1319–33. https://doi.org/10.1016/j.enconman.2019.05.059.

[15] Shamekhi AM, Shamekhi AH. A new approach in improvement of mean value models for spark ignition engines using neural networks. Expert Syst Appl 2015;42:5192–218. https://doi.org/10.1016/j.eswa.2015.02.031.

[16] Li J, Zhou Q, Williams H, Xu H, Du C. Cyber-Physical Data Fusion in Surrogate-assisted Strength Pareto Evolutionary Algorithm for PHEV Energy Management Optimization. IEEE Trans Ind Informatics 2021.

[17] Kocher L, Koeberlein E, Van Alstine DG, Stricker K, Shaver G. Physically based volumetric efficiency model for diesel engines utilizing variable intake valve actuation. Int J Engine Res 2012;13:169–84. https://doi.org/10.1177/1468087411424378.

[18] Yusri IM, Abdul Majeed APP, Mamat R, Ghazali MF, Awad OI, Azmi WH. A review on the application of response surface method and artificial neural network in engine performance and exhaust emissions characteristics in alternative fuel. Renew Sustain Energy Rev 2018;90:665–86. https://doi.org/10.1016/j.rser.2018.03.095.

[19] de Nola F, Giardiello G, Gimelli A, Molteni A, Muccillo M, Picariello R. Volumetric efficiency estimation based on neural networks to reduce the experimental effort in engine base calibration. Fuel 2019;244:31–9. https://doi.org/10.1016/j.fuel.2019.01.182.

[20] Luján JM, Climent H, García-Cuevas LM, Moratal A. Volumetric efficiency modelling of internal combustion engines based on a novel adaptive learning algorithm of artificial neural networks. Appl Therm Eng 2017;123:625–34. https://doi.org/10.1016/j.applthermaleng.2017.05.087.

[21] Li Z, Li J, Zhou Q, Zhang Y, Xu H. Intelligent air/fuel ratio control strategy with a PI-like fuzzy knowledge–

based controller for gasoline direct injection engines. Proc Inst Mech Eng Part D J Automob Eng 2018:095440701877918. https://doi.org/10.1177/0954407018779180.

[22] Li J, Li Z, Zhou Q, Zhang Y, Xu H. Improved scheme of membership function optimisation for fuzzy air-fuel ratio control of GDI engines. IET Intell Transp Syst 2018;13:209–17. https://doi.org/10.1049/iet-its.2018.5013.

[23] Li Z, Zhou Q, Zhang Y, Li J, Xu H. Enhanced intelligent proportional-integral-like fuzzy knowledge–based controller using chaos-enhanced accelerated particle swarm optimization algorithm for transient calibration of air–fuel ratio control system. Proc Inst Mech Eng Part D J Automob Eng 2019:095440701986207. https://doi.org/10.1177/0954407019862079.

[24] Jang JR. ANFIS : Adaptive-Network-Based Fuzzy Inference System. IEEE Trans Syst Man, Cybern Syst 1993;23.

[25] Taghavifar H, Khalilarya S, Jafarmadar S. Adaptive neuro-fuzzy system (ANFIS) based appraisal of accumulated heat from hydrogen-fueled engine. Int J Hydrogen Energy 2015;40:8206–18. https://doi.org/10.1016/j.ijhydene.2015.04.089.

[26] Motahari-Nezhad M, Mazidi MS. An Adaptive Neuro-Fuzzy Inference System (ANFIS) model for prediction of thermal contact conductance between exhaust valve and its seat. Appl Therm Eng 2016;105:613–21. https://doi.org/10.1016/j.applthermaleng.2016.03.054.

[27] Singh NK, Singh Y, Sharma A, Rahim EA. Prediction of performance and emission parameters of Kusum biodiesel based diesel engine using neuro-fuzzy techniques combined with genetic algorithm. Fuel 2020;280:118629. https://doi.org/10.1016/j.fuel.2020.118629.

[28] Zhou Q, Zhao D, Shuai B, Li Y, Williams H, Xu H. Knowledge Implementation and Transfer With an Adaptive Learning Network for Real-Time Power Management of the Plug-in Hybrid Vehicle. IEEE Trans Neural Networks Learn Syst 2021;PP:1–11. https://doi.org/10.1109/TNNLS.2021.3093429.

[29] Zhou Q, Li Y, Zhao D, Li J, Williams H, Xu H, et al. Transferable Representation Modelling for Real-time Energy Management of the Plug-in Hybrid Vehicle based on K-fold Fuzzy Learning and Gaussian Process Regression. Appl Energy 2021.

[30] Wang C, Liang M, Chai Y. Finite-Time Identification Algorithm for Volumetric Efficiency Map in SI Gasoline Engines. IEEE Trans Ind Electron 2020;67:10702–12. https://doi.org/10.1109/TIE.2019.2962481.

[31] Li J, Wu D, Attar M, Xu H. Geometric Neuro-Fuzzy Transfer Learning for In-Cylinder Pressure Modelling of a Diesel Engine Fuelled with Raw Microalgae Oil. Appl Energy 2021.

[32] Zhu J, Ge Z, Song Z, Gao F. Review and big data perspectives on robust data mining approaches for industrial process modeling with outliers and missing data. Annu Rev Control 2018;46:107–33. https://doi.org/10.1016/j.arcontrol.2018.09.003.

[33] Sakthivel G. Prediction of CI engine performance, emission and combustion characteristics using fish oil as a biodiesel at different injection timing using fuzzy logic. Fuel 2016;183:214–29. https://doi.org/10.1016/j.fuel.2016.06.063.

[34] Li J, Zhou Q, Williams H, Xu H. Back-to-back competitive learning mechanism for fuzzy logic based supervisory control system of hybrid electric vehicles. IEEE Trans Ind Electron 2020;67:8900–9. https://doi.org/10.1109/TIE.2019.2946571.

[35] Takagi T, Sugeno M. Fuzzy Identification of Systems and Its Applications to Modeling and Control. IEEE Trans Syst Man Cybern 1985;SMC-15:116–32. https://doi.org/10.1109/TSMC.1985.6313399.

[36] Ojha VK, Snasel V, Abraham A. Multiobjective Programming for Type-2 Hierarchical Fuzzy Inference Trees. IEEE Trans Fuzzy Syst 2018;26:915–36. https://doi.org/10.1109/TFUZZ.2017.2698399.

[37] Chen Y, Yang B, Dong J, Abraham A. Time-series forecasting using flexible neural tree model. Inf Sci (Ny) 2005;174:219–35. https://doi.org/10.1016/j.ins.2004.10.005.

[38] Kennedy J. Particle swarm optimization. Encycl. Mach. Learn., Springer; n.d., p. 760–6.

[39] Hooke R, Jeeves TA. “Direct Search” Solution of Numerical and Statistical Problems. J ACM 1961;8:212–29. https://doi.org/10.1145/321062.321069.

[40] Choi HD, Pak JM, Lim MT, Song MK. A Gaussian distributed resampling algorithm for mitigation of sample impoverishment in particle filters. Int J Control Autom Syst 2015;13:1032–6. https://doi.org/10.1007/s12555-014-0355-2.