

# A comparison of different network based modeling methods for prediction of the torque of a SI engine equipped with variable valve timing

A. H. Kakae\*, B. Mashhadi, M. Ghajar

Associated Professor, Professor, Ph.D. Student Department of Automotive Engineering, Iran university of Science and Technology, Narmak, Tehran, I

\* Corresponding Author

## Abstract

Nowadays, due to increasing the complexity of IC engines, calibration task becomes more severe and the need to use surrogate models for investigating of the engine behavior arises. Accordingly, many black box modeling approaches have been used in this context among which network based models are of the most powerful approaches thanks to their flexible structures. In this paper four network based modeling methods are used and compared to model the behavior of an IC engine: neural networks model (NN), group method of data handling model (GMDH), a hybrid NN and GMDH model (NN-GMDH), and a GMDH model whose structure is determined by genetic algorithm (Genetic-GMDH). The inputs are engine speed, throttle angle, and intake valve opening and closing timing, and the output is the engine brake torque. Results show that NN has the best prediction capability and Genetic-GMDH model has the most flexible and simplest structure and relatively good prediction ability.

**Keywords:** *Neural networks; Group method of data handling; Engine torque; Black box modeling; Variable valve timing*

## 1. Introduction

Nowadays the engine calibration process becomes more complicated and confusing task due to the increasing of the complexity of the engine and its control variables. Usually the calibration engineer has to search the optimum setting for the engine at each speed, and in various operating conditions such as cold start, cruising, etc. So, execution of this task is very time consuming, must trust on the experience of the engineer [1], and may not provide optimum setting for all of the operating conditions. One of the most effective solutions for this problem is black box modeling in which the variations of the torque, fuel consumption, or emissions due to the variations of control variables (such as valve timing, injection timing, etc) are modeled with least prior information about the physics of problem. There are various methods for black box modeling, among which network based methods (such as Neural Networks or Group method of Data Handling) are of most significance. All of these methods mimic the neural structure in living organisms and have an inter-

connected network of neurons (nodes). This modeling schema has a large potential for modeling nonlinear and complicated systems only by using some input-output recorded samples. So these modeling techniques are used widely for modeling complicated phenomena in the fields such as engineering [2-3], chemistry and biochemistry[4], economics [5-6], etc. Specifically, in the field of IC engine, numerous researches have been reported in which these methods have been applied for modeling and prediction of the engine behavior. Ismail et al.[7] used NN for predicting of nine different light duty diesel engine responses, such as carbon monoxide (CO), nitrogen monoxide (NO), maximum pressure, etc. using various blends for the fuel. Results showed that NN models with 10

Neurons and Levenberg-Marquardt training algorithm are the best ones, and seven out of nine engine-out responses can be predicted by these models accurately. Tasdemir et al.[8] applied NN and fuzzy expert system approaches for predicting of the engine torque, emissions, and fuel consumption. Statistical investigations indicated that the both methods have a good ability for data prediction and

there is no significant difference between them. Atashkari et al.[9] used GMDH networks for modeling the engine torque and fuel consumption of a SI engine equipped with VVT. The structure of the model was determined by genetic algorithm. Results showed the superiority of this approach over NN models in data prediction. The developed model was used for variable valve timing optimization. Kalogirou[10] implemented a comprehensive review of the application of such methods in engine and combustion modeling.

Variable valve timing is now one of the well-known technologies capable of increasing the torque and power of the engine, and at the same time reducing its fuel consumption and emissions. In a conventional IC engine the valve timing is fixed and usually is set for optimum performance of the engine at high speeds and wide-open throttle conditions. But this fixed timing is not optimum for all areas of engine operation space. For example at high engine speeds a long overlap period could improve the volumetric efficiency, and hence, the output torque of the engine, but at low speeds this period should decrease in order to prevent the exhaust gas from entering the inlet port excessively. Hence, considerable improvement in the performance and fuel consumption will occur if the variable valve timing technology is used. According to these advantages, during the last few years, many researchers have focused on this technology and its potential improvements[9,11-18].

In this study four network-based modeling methods (namely Neural Networks (NN), Group Method of Data Handling network (GMDH), hybrid neural networks and GMDH algorithm (NN-GMDH), and hybrid genetic GMDH algorithm) are compared in the modeling of the torque of an IC engine in terms of engine speed, throttle position, and inlet valve opening and closing timing. In the following sections first each of these methods is briefly described. Then the method of data collection is described. At last the results of the application of each method in modeling the prepared data are represented and the best method among the others are chosen.

## 2 Network-Based Modeling Methods

As mentioned before, these methods mimic the neural structure in living organisms and have an interconnected network of neurons (nodes). Each node has two or more inputs, a transfer function that transforms these inputs to the output, and probably some weighting coefficients. The outputs of the nodes serve as the inputs to the other nodes, and the output

of the last node is the output of the model. In this way a very complex model can be represented by some processing units (i.e. neurons) interconnecting with each other in a specific manner. This modeling schema has a large potential for modeling nonlinear, complicated, and multidimensional systems only by using some input-output recorded samples. In this paper four types of such models are explained, developed in MATLAB, and their ability to predict the engine behavior are compared.

### 2.1 Artificial Neural Networks

As Nelles stated[19], it is hard to draw a clear line between neural networks and non-neural networks models. The models referred here as NNs are Multi Layer Perceptron (MLP) networks. This is one of the most popular types of NNs models and its description can easily be found in the literature. So it is not explained here.

### 2.2 Group Method of Data Handling (GMDH)

Briefly speaking, GMDH is a self organizing network that its transfer functions are polynomial and its coefficients are determined by regression method. This type of network model first was developed by Ivakhnenko[20], and then enhanced by others[10]. The ability to organize the neurons in a layer and eventually determine the structure of the model is a characteristic distinguishing GMDH from conventional NN models. Furthermore, since the coefficients of each neuron in GMDH network are determined explicitly by regression method, the probability of trapping the solution in local minima (which is usual for a NN training process) is not concerned.

Modeling problem with GMDH networks can be described as follows: Assuming a multi-input single output function  $y=f(x_1, x_2, \dots, x_n)$  with  $m$  samples as  $(x_{i1}, x_{i2}, \dots, x_{in}, y_i) \quad 1 \leq i \leq m$ , the purpose of modeling with GMDH (and any other modeling method) is to produce a function  $y_p = f_p(x_1, x_2, \dots, x_n)$  that for a given input vector, can predict the actual output as precisely as possible. GMDH structure is determined by the number of neuron inputs in each layer ( $p$ ) and the type of polynomial used in the neurons. There are two types of GMDH network, namely basic and modified structure. In the basic type all of the neurons in all layers have the same number of inputs. But in the modified type the number of inputs in each layer can be different. Three types of polynomials usually used in GMDH neurons are linear, quadratic and

modified quadratic. These polynomials with two inputs have the following forms:

$$\text{Linear: } y = a_1x_1 + a_2x_2 + a_3$$

$$\text{Quadratic: } y = a_1x_1^2 + a_2x_2^2 + a_3x_1x_2 + a_4x_1 + a_5x_2 + a_6$$

$$\text{Modified Quadratic: } y = a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4$$

(1) Main steps of the modeling are as follows:

1- In the first layer all of the possible combinations of  $p$  form  $n$  inputs are used to construct neurons. So the number of constructed neurons in the first layer will be:

$$s = \binom{n}{p} \quad (2)$$

2- Using  $m$  samples of the  $m$  samples  $(x_{i1}, x_{i2}, \dots, x_{in}, y_i)$   $1 \leq i \leq m$  as training set, the coefficients of the polynomial of each neuron are determined by a regression method.

3- Of the  $s$  neurons constructed in the first layer, only the ones with best performance survive in the layer and the remaining are discarded. Performance of the neurons are evaluated using the mean square error in the internal test set. This set is the remaining  $rt$  samples ( $rt = m - rtr$ ). This set is merely for choosing the best neurons and should not be confused with test set usually used for performance evaluation of the model after terminating the modeling process, so named here as internal test. In this way the model is prevented from overfitting. The number of survivors in each layer ( $ssurv$ ) is predefined by the user.

4- The outputs of the neurons of the first layer are the inputs for the second layer, and steps 1-3 are repeated for this layer.

5- Steps 1-4 are repeated until the maximum number of layers predefined by the user, or a defined criterion is reached. In the last layer only the neuron with best performance survives and all of the others are discarded. Neurons in all layers having no connection with this neuron are also eliminated.

### 2.3 Genetic GMDH

This modeling method is a modified version of the conventional GMDH method in which the structure of the model is determined by genetic algorithm. The method was first introduced by Narimanzadeh[21], and has shown its capabilities in various problems[21-25]. Modeling procedure is very similar to GMDH modeling approach. But there are two significant differences between classic GMDH and genetic GMDH. First is in the process of structure determination. In genetic GMDH approach, instead of "choosing the best neurons" procedure used in

GMDH, genetic algorithm is applied for structure determination. Second difference refers to the model structure: Here, the inputs of each neuron can come from all of the previous layers. This gives the model much more flexible structure and the chance of having better performance with simpler structure.

For application of genetic algorithm, first the model structure must be encoded so that it can be used as the input variable for the algorithm. This procedure has been shown in Fig. 1 (left) for two typical models. An important point is that if a neuron output passes, without any change, through one layer, its name will be repeated one more time (like the input  $d$  in Fig. 1, bottom left). Accordingly, if a neuron output passes through  $n$  adjacent layers without any change, the number of its repetitions will be  $2n$ .

Using this encoding scheme, the application of genetic algorithm for optimization of the model structure becomes possible. First a function is developed in MATLAB whose input is the code of a structure, and its output is the cost function of the genetic algorithm. This cost function is defined here as the root mean square error of the model on the training data:

$$E = \sqrt{\frac{\sum_{i=1}^n (y_i - y_{pi})^2}{n}} \quad (3)$$

Which genetic algorithm tries to minimize. First a population of several structures is randomly generated. Then this population evolves by means of genetic operators such as crossover and mutation, and finally the best chromosome in the last generation is chosen as the best structure.

Using genetic algorithm and its operators (crossover and mutation) with the described encoding scheme is straightforward. Fig. 1 shows crossover operation for two chromosomes. For this operation a point in the both chromosomes is randomly chosen. This point should be chosen only from the set  $\{21, 22, 23, \dots, 2nl\}$  where  $nl$  is the number of hidden layers in the smaller chromosome.

Mutation operation is also easily executed by randomly choosing a gene in a chromosome and replacing it randomly to another legal alphabetical character.

In this way, by means of GMDH approach to find the coefficients of the model, and genetic algorithm to find the best structure for the model, a more flexible and powerful modeling scheme is developed in comparison with conventional NN and GMDH modeling methods.

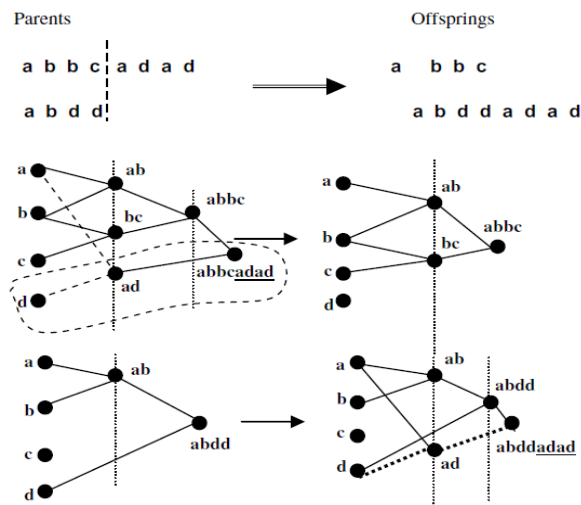


Fig1. Two typical models for representing the encoding scheme and crossover operation. Each letter denotes an input [9]

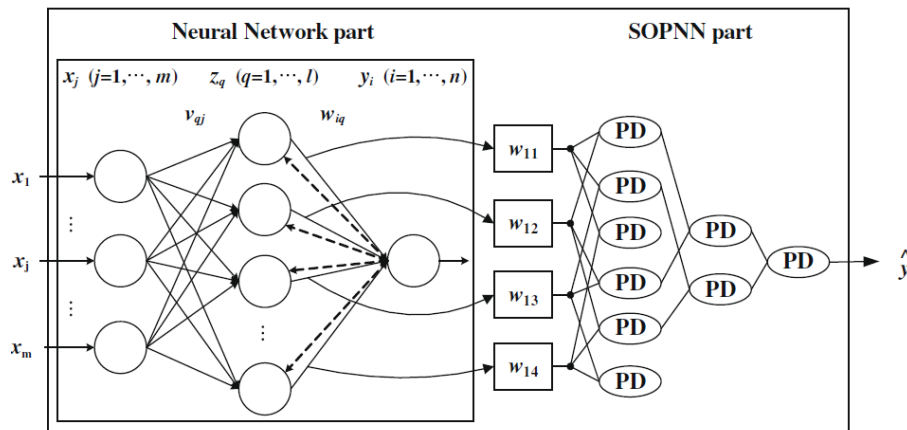


Fig2. Schematic diagram of a hybrid NN GMDH model[25] (SOPNN stands for Self Organizing Polynomial Neural Network another name for GMDH networks.)

Table 1 Specifications of the engine used for modeling

Engine type	Four Stroke, SI
Aspiration	Normal
Fuel System	Multi-port Fuel Injection
Number of cylinders	4
Number of valves	8
Displacement (cc)	1761
Bore (mm)	83
Stroke (mm)	81.4
Compression Ratio	9.3:1
Max I/O Valve Lift (mm)	9.7

## 2.4 Hybrid NN-GMDH

This method, firstly introduced by Kim et al.<sup>[25]</sup>, is a combination of NN and GMDH models in a cascade form. The model consists of two parts. In the first part a conventional neural network and in the second part, a GMDH network is developed, and the outputs of each neuron in the last hidden layer of the NN serve as the inputs to the GMDH model. Training of the NN and GMDH networks is implemented conventionally. Fig. 2 shows this hybrid model schematically.

According to the author, this hybrid model provides a synergistic environment and can handle the limitation of GMDH method in modeling the systems with few inputs. So, it was chosen here as a candidate method for modeling.

## 3 Data Gathering and Preparation

For data gathering one must run an actual engine on the dynamometer and record the inputs and outputs. But this method can be expensive and time consuming, specifically if the number of data points is large, or adjusting the input variables is hard to implement. So, we use a comprehensive numerical model of the actual engine. This model should have acceptable accuracy and computational cost. There are various models for engine simulation such as zero, one, two, and three dimensional models, with various levels of accuracy and computational cost. Among the others, one dimensional models are able to compromise between accuracy and computational cost in a satisfactory manner for this study. So, a comprehensive one dimensional model was developed in GT POWER in which main phenomena such as gas flow through inlet and outlet valves, and combustion, along with all the engine components such as inlet and outlet manifolds, catalyst convertor, filter, muffler, and exhaust have been carefully modeled. Combustion was modeled with wiebe function and its coefficients were determined based on the experimental data recorded from actual engine. The engine type is XU7/JP4/L3 and its specifications have been shown in Table 1.

The accuracy of this model for predicting the engine brake power is shown in Fig. 3. As can be seen, the developed engine model has a good accuracy except for high engine speeds. Maximum and mean relative error are 11% and 4% respectively.

Latin Hypercube method, being more suitable than other methods such as factorial design for

numerical experiments[26-27] is used here for designing the matrix of experiments. This involves the definition of range of each variable, and the number of divisions for them. The ranges of the variables are shown in Table 2. Each variable subdivides into 78 intervals with equal probability.

From the 78 samples the first 63 samples are used for training and the last 15 ones are used for testing the ability of the models in predicting of the unseen data.

## 4 Results and discussion

### 4.1 Artificial Neural Networks

Of the 63 samples for training, 57 samples are used explicitly for training the network, and the remaining 6 samples are used for early stopping, a method for preventing the model from overfitting the data. Since the best structure for the data in hand is not known initially, a process of trial and error should be accomplished in order to find the best structure. Accordingly, structures with one hidden layer and 5, 10, and 15 neurons have been examined. Two types of transfer function also have been tried. Hyperbolic tangent (tansig), having saturation behavior, and linear function (purelin). In order to comparison the performance of different models in a quantitatively manner, three criterions have been used: Root mean square error (RMSE), max error (MAXE), and absolute fraction of variance (R2) defined as:

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{\sum_{i=1}^n (y_i - y_{pi})^2}{n}} \\ \text{MAXE} &= \max(y_i - y_{pi}) \quad 1 \leq i \leq n \\ R^2 &= 1 - \frac{\sum_{i=1}^n (y_i - y_{pi})^2}{\sum_{i=1}^n (y_i)^2} \end{aligned} \quad (4)$$

Where n denotes the number of training or test data, and  $y_i$ ,  $y_{pi}$  are actual and estimated outputs respectively. Results have been shown in the Table A.1 in appendix A. For an easy comparison, the RMSE of the models has been represented graphically in Fig. 4.

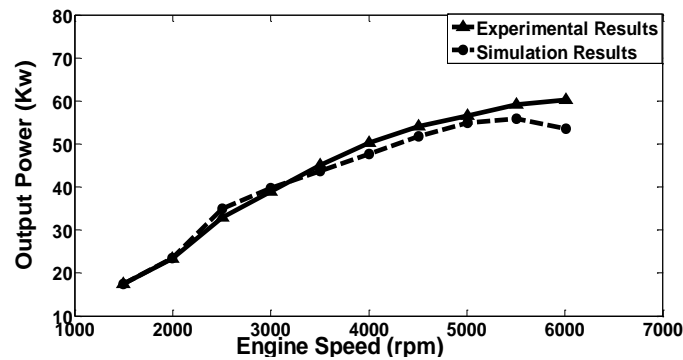


Fig3. Actual engine and simulated model brake power at different engine speeds.

Table 2 Selected range for each variable.

Input Variable	Minimum	Maximum
Inlet valve-opening timing (Deg, aTDC)	-40	10
Inlet valve-closing timing (Deg, aBDC)	0	80
Throttle Position (Deg)	0	90
Engine Speed (rpm)	1000	6000

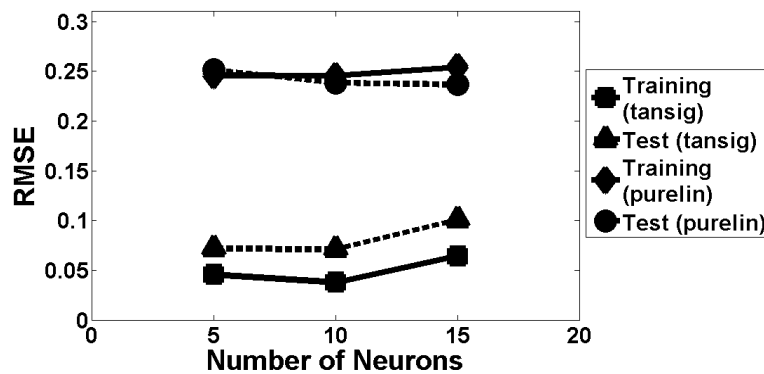


Fig4. RMSE values for various NN models.

As can be seen, when tansig transfer functions are used, the test error gradually increases with increasing the neurons, but training error first slightly reduces and then increases. It must be noted that early stopping prevents training error from becoming very small, otherwise training error would reduce monotonically by increasing the number of neurons.

The model with 15 neurons has the biggest test error which is a sign of excessive complexity of the model. The models with 5 or 10 neurons have reasonable training and test error, and so can be chosen as the best model.

All models with linear transfer functions have nearly the same training and test errors. In addition,

the values of these two errors are close to each other. The reason refers to the linear transfer functions which make the whole model be linear, independent of the number of neurons. As can be seen, linear transfer functions causes much more error in comparison with tansig functions and are not suitable for the data in hand. This reflects the nonlinear behavior of the system. So the best model is the one having 5 or 10 neurons with tansig functions.

#### 4.2 Group Method of Data Handling (GMDH) Networks

Of the 63 samples for training 50 samples are used explicitly for training, and the 13 remaining samples for choosing the best neurons in each layer. Here, we use structures with 2, 3, and 4 layers, and quadratic transfer functions for all of the neurons. Both basic and modified networks are developed. In the basic form the neurons in all layers have two inputs. In the modified form the neurons of the first layer have 2 inputs and in the other layers have 3 inputs. Also the number of the remaining neurons in each layer is set to 30. Results have been shown in the Table A.2 in appendix A. The RMSE of the models has been represented graphically in Fig. 5.

According to the results, in modified form, training error reduces gradually with increasing the number of layers. Test error also firstly reduces, but increases afterward. This shows that the model with 4 layers has excessive complexity resulting in overfitting the data. In basic models, increasing the

layers doesn't have any meaningful effect on the training and test errors except for one case in which the training error decreased by increasing the layers from 2 to 3. The other point is that modified models usually have lower training and test errors in comparison with basic models. It can be concluded that modified model with 3 layers is the best one having reasonable training and test errors. In Fig. 6 the structure of the this model has been shown.

#### 4.3 Genetic GMDH models

Here, the major parameters affecting the training process refer to the genetic algorithm. For this algorithm a population of 32 individuals is made and evolved by the crossover and mutation operators in order to make the next generations. Mutation rate is set to 0.15. 30% of the all individuals in a generation are directly transferred to the next generation and total number of generations is 200. Here, in order to choose models with more prediction power, training data set is subdivided into two subsets: internal training data including 50 samples and used to determine the coefficients of the model, and internal test data including 13 samples used to evaluate the cost function for the model. Results for the best models with 3, 4, 5, and 6 layers have been represented in Table A.3.

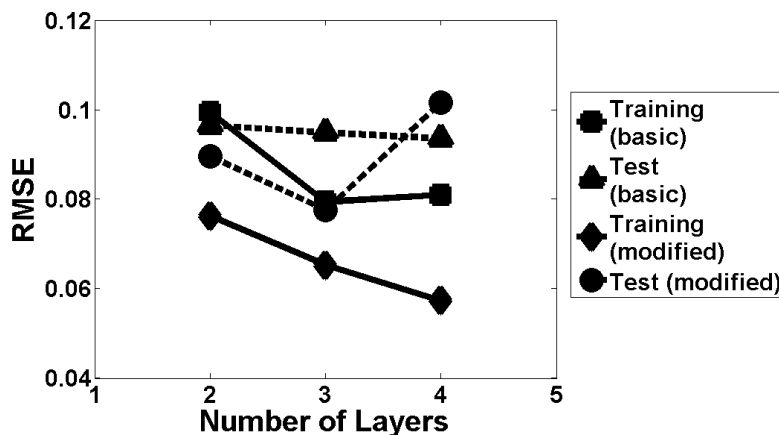


Fig5. RMSE values for various GMDH models

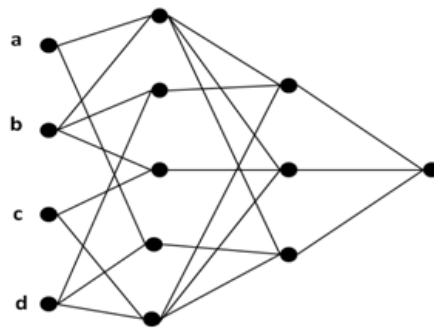


Fig6. The structure of the selected GMDH model.

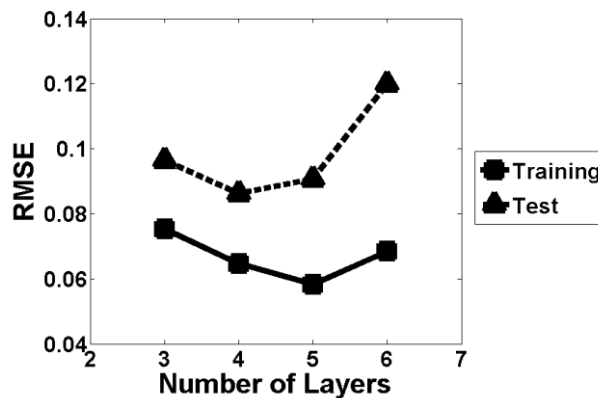


Fig7. RMSE values for various Genetic GMDH models.

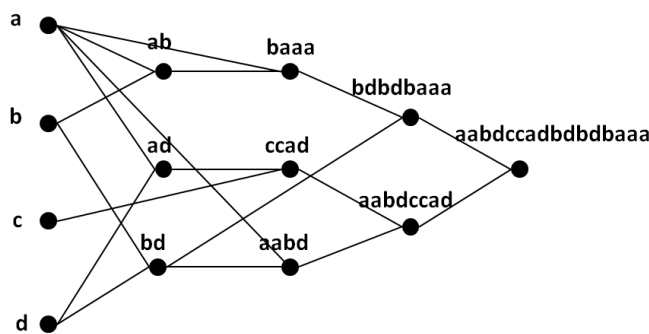


Fig8. The structure of the selected Genetic GMDH model.

Root mean square errors of the models on training and test data sets have been shown in Fig. 7. Results shows that the both errors first reduce and then increase with increasing the layers. Training error is minimum in the model with 5 layers, and test error in the model with 4 layers. Model with 3 layers has relatively high training and test errors which may be due to its simple structure and excluding input 3 from the model. Also the model with 6 layers has a large test error reflecting the extra complexity of the model. It can be concluded that the model with 4 layers is the best one compromising among training and test errors and complexity of the model in the best way. The structure of this model has been represented in Fig. 8.

#### 4.4 Hybrid NN-GMDH models

The NN part has 10 neurons and both tangent hyperbolic and linear functions are used for neurons. Furthermore, its training parameters are the same as the NN model developed before. Likewise, the GMDH part is very similar to the GMDH model developed in section 4.2. Both basic and modified type networks with 2, 3, and 4 layers are tried. Results have been shown in Table A.4 and A.5 in appendix A, and Fig. 9. Based on the results, test errors of the NN-GMDH models are usually higher than the corresponding errors in other types of model. Errors



don't have regular trends when tangent hyperbolic transfer functions are used in NN part. Also in most cases the training and test errors of the NN part are less than the corresponding errors of the whole model. This may be due to the saturation behavior of tangent hyperbolic function. This behavior causes the majority of the components of input vectors to the GMDH part to lie near 1 or -1 which in turn results in ill-conditioning. In addition, as stated before, the least squares method itself is susceptible to ill-conditioning. These can cause the training, and specifically test errors, to become relatively high, especially if the model is of modified type with 4 layers (i.e. has more complexity). Fig. 10 shows prediction of the test set by a modified model with 4 layers that suffers from this problem. However, as linear functions don't saturate, this is not the case when linear transfer functions are used in NN part. In both basic and modified types training error reduces monotonically as the layers increase, and test error has a minimum in models with 3 layers. Hence the modified model with 4 layers and linear transfer functions in NN part can be chosen as the best model among the others. The structure of this model has been shown in Fig. 11.

#### 4.5 Comparison among different methods

Root mean square and maximum error of the chosen models, on the training and test data sets, have been represented in Fig. 12 and 13 respectively.

As can be seen NN model with 10 neurons has the lowest training and test error, and hence the best prediction power among the others. Furthermore, GMDH and Genetic GMDH models have nearly the same results, and hybrid NN-GMDH model has the least prediction capability among the others. Maximum errors also reflect the same facts.

The other comparable characteristic is the complexity of the models. Total number of neurons in

the model and the sum of the number of the inputs of all neurons are the two criteria chosen here for evaluating the complexity. Fig. 14 shows these two quantities for the chosen models.

Results shows that genetic GMDH model with 9 neurons and 18 inputs has the least complexity, on the other hand NN-GMDH with 21 neurons and 66 inputs is the most complex model. Also, the NN has 10 neurons and 40 inputs, and GMDH has 9 neurons and 22 inputs.

Other noteworthy characteristic is the total training time of each model. Results for the chosen models have been shown in Table 3. It can be seen that GMDH model has the lowest training time. The relatively long training time of Genetic GMDH models is due to the nature of the genetic algorithm that involves several generations each consisting of many individuals (i.e. models). As mentioned before, the number of generations is 200, and there are 32 individuals in each generation. So the average time to for implementing the genetic algorithm and training for each individual is approximately 0.0009 Sec. The long training time in this method is the cost of finding the best structure for the model.

It can be concluded that NN model has the best prediction capability and Genetic GMDH model has the simplest structure. GMDH model also has a good prediction capability and the least training time. However, NN model has no self organizing properties. But GMDH and Genetic GMDH models can train the data and determine the appropriate structure simultaneously. In this regard Genetic GMDH models are more powerful due to more flexible structures. Hence, NN and Genetic GMDH models can be chosen as the best models, the first one for its prediction power, and the second one for its self organizing characteristic and flexible structure. The performance of the NN model with 10 neurons has been shown in Fig. 15 and 16.

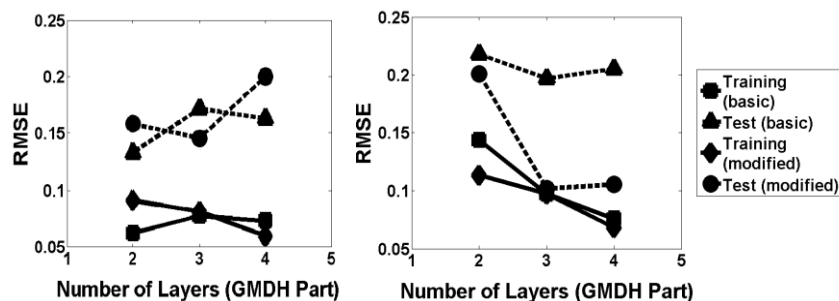


Fig9. RMSE values for various NN-GMDH models: Tangent hyperbolic functions in NN part (left), linear functions in NN part (right).

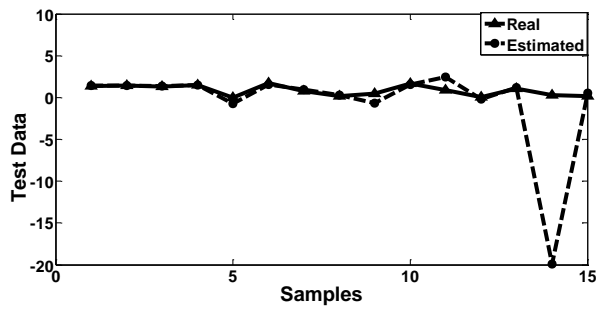


Fig10. An example of weak prediction of the test data with a modified NN-GMDH model with 4 layers. RMSE on test data is 5.21.

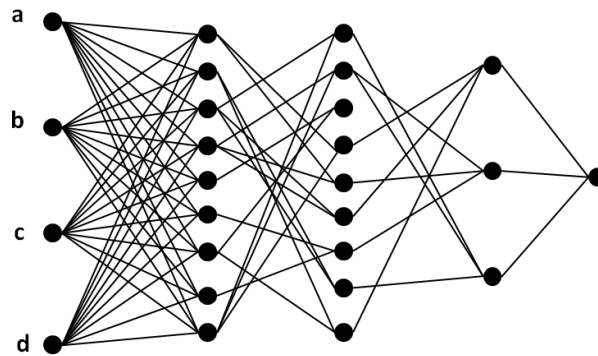


Fig11. The structure of the selected NN-GMDH model.

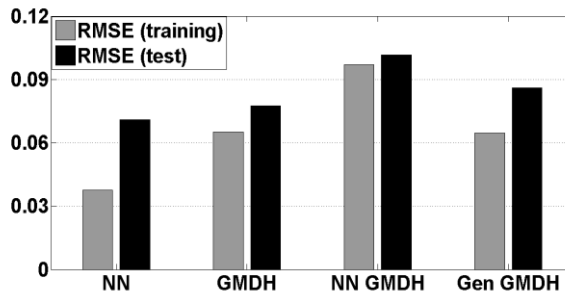


Fig12. Root mean square error on the training and test data sets for the chosen models.

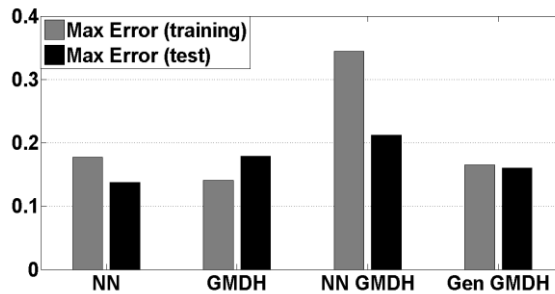
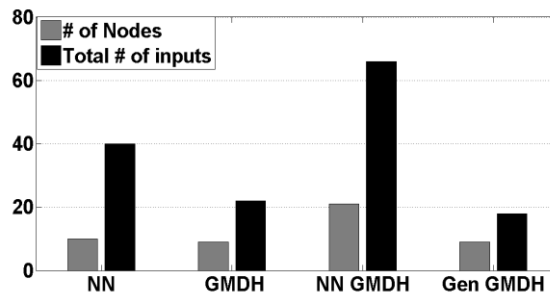


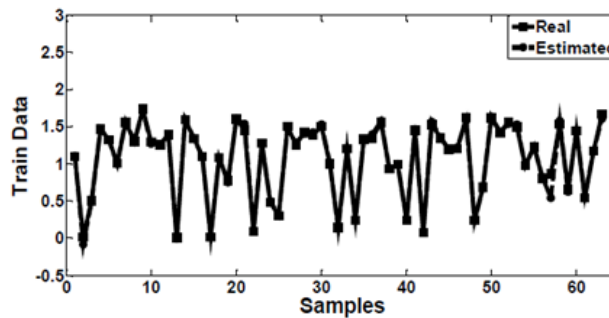
Fig13. Maximum error on the training and test data sets for the chosen models.



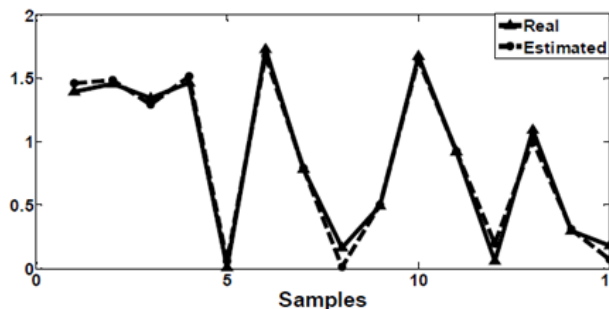
**Fig14.** A comparison of the complexity of the chosen models in terms of total number of neurons in the model and the sum of the number of their inputs.

**Table 3** Training time of various methods on a computer with intel® core i7-2670QM CPU and 8 GB of RAM.

NN	0.70
GMDH	0.26
NN-GMDH	3.07
Genetic-GMDH	6.30



**Fig15.** Actual and estimated outputs of the training data by NN model with one hidden layer, 10 neurons, and tansig transfer functions.



**Fig16.** Actual and estimated outputs of the test data by NN model with one hidden layer, 10 neurons, and tansig transfer functions.

### 5 conclusions

In this paper four network based modeling approaches were explained, used and compared to model the behavior of an IC engine with VVT: neural networks model (NN), group method of data handling

model (GMDH), a hybrid NN and GMDH model (NN-GMDH), and a GMDH model whose structure is determined by genetic algorithm (Genetic-GMDH). Engine speed, throttle angle, and inlet valve opening and closing timing were the inputs, and the output

was the engine brake torque. Comparison among these models implemented based on root mean square and maximum error of the models on the training and test data sets, the complexity of the models, and the training time. Results showed that NN model has the best prediction ability, and

Genetic-GMDH model has the simplest and most flexible structure, and reasonable prediction power.

However the results of the GMDH and the Genetic-GMDH models were also close to each other. Furthermore, the GMDH model training time was much less than the genetic GMDH training time. Hence these approaches can be used to developed surrogate models for more complex systems with more inputs. NN-GMDH model didn't have satisfactory results in terms of training and test error, model complexity, and training time.

## References

- [1]. VONG CM, WONG PK, LI YP. Prediction of automotive engine power and torque using least squares support vector machines and Bayesian inference [J]. *Engineering Applications of Artificial Intelligence*, 2006, 19: 277-287.
- [2]. LI F, ZHANG J, OKO E, et al. Modelling of a post-combustion CO<sub>2</sub> capture process using neural networks [J]. *Fuel*, 2015, 151: 156-163.
- [3]. Xiao B, Wang S, and Prucka RG. An investigation of semiphysical artificial neural networks for multi-fuel combustion phasing control of spark ignition engines [J]. *ImechE Part D: Automobile Engineering*, 2014, 228(11): 1262-1273.
- [4]. DORN M, BRAGA ALS, LLANOS CH, et al. A GMDH polynomial neural network-based method to predict approximate three-dimensional structures of polypeptides [J]. *Expert Systems with Applications*, 2012, 39: 12268–12279.
- [5]. FALAT L, PANCIKOVA L. Quantitative Modelling in Economics with Advanced Artificial Neural Networks [J]. *Procedia Economics and Finance*, 2015, 34: 194-201.
- [6]. VARAHRAMI V. Good Prediction of Gas Price between MLFF and GMDH Neural Network [J]. *International International Journal of Finance and Accounting*, 2012,1(3): 23-27.
- [7]. ISMAIL HM, NG HK, QUECK CW, et al. Artificial neural networks modelling of engine-out responses for a light-duty diesel engine fuelled with biodiesel blends [J]. *Applied Energy*, 2012, 92: 769–777.
- [8]. TASDEMIR S, SARITAS I, CINIVIZ M, et al. Artificial neural network and fuzzy expert system comparison for prediction of performance and emission parameters on a gasoline engine [J]. *Expert Systems with Applications*, 2011, 38: 13912-13923.
- [9]. [9] ATASHKARI K, NARIMAN-ZADEH N, GOLCU M, et al. Modelling and multi-objective optimization of a variable valve-timing spark-ignition engine using polynomial neural networks and evolutionary algorithms [J]. *Energy Conversion and Management*, 2007, 48: 1029-1041.
- [10]. KALOGIROU SA. Artificial intelligence for the modeling and control of combustion processes: a review [J]. *Progress in Energy and Combustion Science*, 2003, 29: 515-566.
- [11]. BOHAC SV, ASSANIS DN, Effect of Exhaust valve Timing on Gasoline engine Performance and Hydrocarbon Emissions [C]. *SAE Paper*, 2004-01-3058, 2004.
- [12]. FONTANA G, GALLONI E. Variable valve timing for fuel economy improvement in a small spark ignition engine [j]. *Applied Energy*, 2009, 86: 96-105.
- [13]. JIA M, XIE M, WANG T, et al. The effect of injection timing and intake valve close timing on performance and emissions of diesel PCCI engine with a full engine cycle CFD simulation [J]. *Applied Energy* 2011, 88: 2967-2975.
- [14]. LEROY T, CHAUVIN J, PETIT N. Motion planning for experimental air path control of a variable-valve-timing spark ignition engine [J]. *Control Engineering Practice*, 2009, 17(12):1432-1439.
- [15]. KOCHER L, KOEBERLEIN E, VAN ALSTINE DG, et al. Physically based volumetric efficiency model for diesel engines utilizing variable intake valve actuation, [J]. *International Journal of Engine Research* 2011, 13: 169-184.
- [16]. KOHANY T, SHER E. Using the 2nd Law of thermodynamics to optimize variable valve-timing for maximizing torque in a throttled SI engine [C]. *SAE paper*, 1999-01-0328, 1999.
- [17]. SHAYLER P, ALGER L. Experimental investigations of intake and exhaust valve timing effects on charge dilution by residuals, fuel consumption and emissions [C]. *SAE paper*, 2007-01-0478, 2007.
- [18]. WU B, PRUCKA RG, FILIPI ZS, et al. Cam Phasing Optimization Using Artificial Neural Networks as Surrogate Model-Maximizing

- output Torque [C]. SAE Paper, 2005-01-3757, 2005.
- [19]. NELLES O. Nonlinear system identification: From classical approaches to neural networks and fuzzy models [M]. Springer, Berlin, 2001.
- [20]. IVAKHENKO AG. Polynomial theory of complex systems [C]. IEEE Transaction on Systems, Man and Cybernetics. SMC-12, IEEE Press, 1971: 364–378.
- [21]. NARIMAN-ZADEH N, DARVIZEH A, AHMAD-ZADEH GR. Hybrid genetic design of GMDH-type neural networks using singular value decomposition for modelling and prediction of the explosive cutting process [J]. Journal of Engineering Manufacture, 2003, 217: 779-790.
- [22]. HAMID-ZADEH N, JAMALI A, NARIMAN-ZADEH N, et al. A Polynomial Model for Concrete Compressive Strength Prediction using GMDH-type Neural Networks and Genetic Algorithm [C]. Proc. of 5th WSEAS International Conference on System Science and Simulation Engineering, Tenerife, Canary Islands, Spain, 2006: 13-18.
- [23]. MADANDOUST R, GHAVIDEL R, NARIMAN-ZADEH N. Evolutionary design of generalized GMDH-type neural network for prediction of concrete compressive strength using UPV [J]. Computational Materials Science, 2010, 49: 556-567.
- [24]. JAMALI A, NARIMAN-ZADEH N, DARVIZEH A, et al. Multi-objective evolutionary optimization of polynomial neural networks for modelling and prediction of explosive cutting process [J]. Engineering Applications of Artificial Intelligence, 2009, 22: 676-687.
- [25]. KIM D, SEO SJ, PARK GT. Hybrid GMDH-type modeling for nonlinear systems: Synergism to intelligent identification [J]. Advances in Engineering Software, 2009, 40: 1087-1094.
- [26]. SALTELLI A, CHAN K, SCOTT E M. (Editors), Sensitivity Analysis [M]. Chichester, John Wiley & Sons, 2000.
- [27]. MCKAY MD, BECKMAN RJ, CONOVER WJ. A comparison of three methods of selecting values of input variables in the analysis of output from a computer code [J]. Technometrics, 1979, 21: 239-245.

**APPENDIX A**

**Table A.1** Results of the prediction of training and test data sets by various NN models.

Transfer Function	# of Neurons	RMSE	MAXE	R2
tansig	5	Train=0.0456 Test=0.0716	Train=0.1405 Test=0.1586	Train=0.9985 Test=0.9953
	10	Train=0.0377 Test=0.0711	Train=0.1771 Test=0.1378	Train=0.9990 Test=0.9954
	15	Train=0.0468 Test=0.0718	Train=0.1238 Test=0.1837	Train=0.9984 Test=0.9955
purelin	5	Train=0.2459 Test=0.2513	Train=0.6160 Test=0.4274	Train=0.9542 Test=0.9402
	10	Train=0.2455 Test=0.2383	Train=0.6348 Test=0.3853	Train=0.9548 Test=0.9466
	15	Train=0.2542 Test=0.2364	Train=0.7831 Test=0.3774	Train=0.9534 Test=0.9483

**Table A.2** Results of the prediction of training and test data sets by various GMDH models.

Type	# of Layers	RMSE	MAXE	R2
Basic	2	Train=0.0994 Test=0.0965	Train=0.2544 Test=0.2260	Train=0.9928 Test=0.9914
	3	Train=0.0794 Test=0.0949	Train=0.2005 Test=0.1909	Train=0.9955 Test=0.9916
	4	Train=0.0810 Test=0.0936	Train=0.1712 Test=0.2056	Train=0.9953 Test=0.9920
Modified	2	Train=0.0763 Test=0.0896	Train=0.1568 Test=0.2043	Train=0.9958 Test=0.9926
	3	Train= 0.0652 Test= 0.0775	Train=0.1411 Test=0.1788	Train=0.9970 Test=0.9945
	4	Train=0.0573 Test=0.1017	Train=0.1388 Test=0.1960	Train=0.9977 Test=0.9905

**Table A.3** Results of the prediction of training and test data sets by various genetic GMDH models.

# of Layers	RMSE	MAXE	R2
3	Train=0.0753 Test=0.0964	Train=0.1658 Test=0.2166	Train=0.9960 Test=0.9914
4	Train=0.0646 Test=0.0862	Train=0.1654 Test=0.1600	Train=0.9970 Test=0.9934
5	Train=0.0582 Test=0.0907	Train=0.1554 Test=0.2196	Train=0.9976 Test=0.9927
6	Train=0.0684 Test=0.1199	Train=0.1534 Test=0.2861	Train=0.9966 Test=0.9859

**Table A.4** Results of the prediction of training and test data sets by various NN-GMDH models with tansig transfer functions.

Type	# of Layers	RMSE (NN)	RMSE	MAXE	R2
Basic	2	Train=0.0462 Test=0.0911	Train=0.0621 Test=0.1329	Train=0.2127 Test=0.2445	Train=0.9972 Test=0.9843
	3	Train=0.0424 Test=0.0679	Train=0.0774 Test=0.1714	Train=0.2249 Test= 0.3681	Train=0.9956 Test=0.9757
	4	Train=0.0694 Test=0.1654	Train=0.0729 Test=0.1629	Train=0.1777 Test=0.2980	Train=0.9961 Test=0.9774
Modified	2	Train=0.0502 Test=0.1408	Train= 0.0908 Test=0.1583	Train=0.3148 Test=0.3098	Train=0.9940 Test= 0.9778
	3	Train=0.0603 Test=0.0914	Train=0.0810 Test=0.1454	Train=0.3440 Test=0.2694	Train=0.9952 Test=0.9810
	4	Train=0.0624 Test=0.1608	Train=0.0594 Test=0.2004	Train=0.2093 Test=0.5111	Train=0.9974 Test=0.9643

**Table A.5** Results of the prediction of training and test data sets by various NN-GMDH models with purelin transfer functions.

Type	# of Layers	RMSE (NN)	RMSE	MAXE	R2
Basic	2	Train=0.2463 Test=0.2435	Train=0.1439 Test=0.2179	Train=0.3656 Test=0.5047	Train=0.9848 Test=0.9603
	3	Train=0.2479 Test=0.2211	Train=0.0973 Test=0.1970	Train=0.2574 Test=0.4153	Train=0.9931 Test=0.9683
	4	Train=0.2473 Test=0.2289	Train=0.0759 Test=0.2052	Train=0.1698 Test=0.4529	Train=0.9958 Test=0.9639
Modified	2	Train=0.2504 Test=0.2859	Train=0.1134 Test=0.2010	Train=0.2697 Test=0.3497	Train=0.9906 Test=0.9619
	3	Train=0.2484 Test=0.2213	Train=0.0971 Test=0.1018	Train=0.3447 Test=0.2120	Train=0.9931 Test=0.9906
	4	Train=0.2467 Test=0.2501	Train=0.0680 Test=0.1055	Train=0.1447 Test=0.1846	Train=0.9966 Test=0.9901

Nomenclature					
NN	Neural Networks		RMSE	Root Mean Square Error	
GMDH	Group Method of Data Handling		MAXE	Maximum Error	
VVT	Variable Valve Timing		aTDC	After Top Dead Center	
SI	Spark Ignition		aBDC	After Bottom Dead Center	
IC	Internal Combustion				