Advanced, Guided Procedure for the Calibration and Generalization of Neural Network-Based Models of Combustion and Knock Indexes

*Published Version:*
Brusa A., Shethia F.P., Mecagni J., Cavina N. (2023). Advanced, Guided Procedure for the Calibration and Generalization of Neural Network-Based Models of Combustion and Knock Indexes. SAE INTERNATIONAL JOURNAL OF ENGINES, 17(2), 153-164 [10.4271/03-17-02-0009].

*Availability:*
This version is available at: https://hdl.handle.net/11585/949533 since: 2024-05-14

*Published:*

DOI: http://doi.org/10.4271/03-17-02-0009

# Advanced, Guided Procedure for the Calibration and Generalization of Neural Network-based Models of Combustion and Knock Indexes

## Abstract

In the last few years, the artificial neural networks have been widely used in the field of the engine modelling. Some of the main reasons for this being, their compatibility with the real-time systems, higher accuracy and flexibility if compared to other data-driven approaches. One of the main difficulties using this approach is the calibration of the network itself. It is very difficult to find in the literature procedures that guide the user to completely define a network. Typically, the very last steps (like the choice of the number of neurons) has to be selected by the user on the base of his sensitivity to the problem.

This work proposes an automatic calibration procedure for the artificial neural networks, considering all the main hyper-parameters of the network such as the training algorithms, the activation functions, the number of the neurons, the number of epochs, and the number of hidden layers, for modelling various combustion indexes in a modern internal combustion engine. However, the proposed procedure can be applied to the training of any neural network-based model.

The automatic calibration procedure outputs a configuration of the network, giving the optimal combination in terms of hyper-parameters. The decision of the optimal configuration of the neural network is based on a self-developed formula, which gives a rank of all the possible hyper-parameter combination using some statistical parameters obtained comparing the simulated and the experimental values. At the end, the lowest rank is selected as the optimal one as it represents the combination having the lowest error. Following the definition of this rank, high accuracy on the results has been achieved in terms of the Root Mean Square Error index, for example, on the combustion phase model, the error is 0.139° crank angle under steady state conditions. On the maximum in-cylinder pressure model, the error is 1.682 Bar, while the knock model has an error of 0.457 Bar for the same test that cover the whole engine operating field.

## Introduction

With the goal of becoming climate neutral by 2050, the European Emission norms are becoming more and more stringent for the transportation sector. However, the studies carried out by important manufacturers like Porsche, have shown that the early exit of Internal Combustion Engines from the market would lead to an increase in $CO_2$ emissions [1]. The virtual sensing of the main combustion indexes and the pollution concentration will have a key role in the onboard monitoring strategies required by the Euro7 regulations. To comply with this goal, one of the important steps is to have solutions that can accurately predict and control the engine parameters in Real-Time (RT). Having an accurate combustion model helps in more precise control of the various actuations available to control the engine thus improving the efficiency [2,3]. Today the state-of-the-art for the combustion models is composed by the following categories:

- Three-dimensional (3D) Computational Fluid Dynamic (CFD) models [4,5].
- One-dimensional (1D) physical models [6,7].
- Zero-dimensional (0D) models [8,9].

The 3D CFD models require high computational power, making them incompatible with the implementation in RT systems. The 1D models could be further simplified and can be made compatible with RT systems using some commercial software [10,11]. The 0D models are based on simplified physical equations, the Mean-Value Models, or the machine learning algorithms such as the Artificial Neural Networks (ANN). The ANN models are typically used for the RT systems, and they require less computational power with respect to the 3D and the 1D models [12]. Literature has shown that for the same available computational power and hardware, the ANN models have similar or in some cases better performance than the conventional models [13].

ANNs have gained a lot of attention in recent years in a wide range of applications, including internal combustion engines. Today the RT systems play a vital role in increasing the efficiency and the performance of the engines [14,15]. Thanks to these systems, the control parameters can be monitored and adjusted in real-time allowing the system to operate at the maximum efficiency. Because of their structure and computational requirements, the ANNs are being widely used in these systems.

Even though ANNs have many advantages over the analytical approaches, there are still some critical concerns, for e.g., the calibration of the network itself which is one of the major challenges faced while using this approach, ANNs are data-driven models, and they require a good quantity of data to give the required results, which might not be always available.

This work proposes an automatic calibration algorithm together with a novel accuracy index, as a solution to improve the network calibration, enhancing the reliability and the accuracy of the resulting model. This work focuses on the supervised Feedforward Neural Networks (FNN) for modelling the main combustion and knock indexes such as the combustion phase (evaluated as the 50% of mass fraction burned, named MFB50), the maximum in-cylinder pressure (Pmax), the Indicated Mean Effective Pressure (IMEP), a statistical index of Maximum Amplitude of Pressure Oscillation (in particular the 98th percentile, indicated as MAPO98). The inputs to these models are the variables/actuations that are typically available onboard and that characterize the operating point such as the engine speed and load, the Spark Advance (SA), and lambda ($\lambda$).

The first part of the paper outlines the experimental campaign followed to collect data to be used in calibrating and developing the models. The experimental tests have been carried out to cover a large part of the engine operating field.

The second part of the paper focuses on the development of the automatic procedure for calibrating the ANNs for various combustion indexes mentioned earlier. The calibration algorithm has been developed in Matlab along with a custom Graphical User Interface (GUI) to calibrate the ANNs. To summarize, this procedure carries out a sweep of different hyper-parameters such as the training algorithms, the activation functions, and the number of neurons for the given dataset, and outputs the optimal values for all such parameters that could be used for modelling a particular combustion index. During the calibration process, the simulated results are compared with the experimental ones and the performances are evaluated using the Root Mean Squared Error (RMSE) index on the test dataset. Then a novel accuracy index “Rank” that weighs the statistical quantities is calculated, and the lowest rank is considered the optimal one, as it represents the combination having the lowest error. This index can be applied for evaluating the accuracy of a general ANN. In this work, only shallow neural networks (networks with a single hidden layer, one input layer and one output layer) have been considered because, from the literature it is seen that this configuration is most commonly used for this type of application [19-21] and furthermore, some tests are carried out with multiple hidden layers which did not yield the results with acceptable accuracy. The number of epochs was set to 8000, from the previous work and experience of the authors [16].

The most innovative contribution is the guided procedure to completely define the network by driving the user to identify the best trade-off between the generalization of model and the punctual accuracy. The standard procedures leave such decision to the sensitivity and experience of the user even if the mentioned trade-off can change moving between different models. The proposed procedure offers more quantitative details to finally define the features of the network.

The main focus of this work is on the development and calibration of an ANN and hence the models have not been tested in a RT system. In a previous work by authors, a similar ANN architecture has been tested in a RT system, demonstrating the performance of an ANN is such a system [22].

## Experimental Campaign

The experimental activity was carried out on a 1.4L, inline 4-cylinder, Turbo-Charged, Gasoline Direct Injection engine installed in the test cell, whose main characteristics are listed in the Table 1. The engine is equipped with a piezoelectric pressure sensor for each cylinder. This pressure sensor is from Kistler, and the main features of which are reported in Table 2. Data from these sensors are recorded with a sampling frequency of 200kHz. The Alma Automotive mASTRO charge amplifier and the OBI indicating system (provided by the same manufacturer) are applied for this activity. All the other sensors installed on the engine (thermocouples and the piezoresistive sensors) are sampled with a frequency of 100 Hz by the test bench software.

**Table 1**. Engine characteristics.

| | |
|---|---|
| Displaced volume | 1389.9 cc |
| Stroke | 75.6 mm |
| Bore | 76.5 mm |
| Connecting Rod | 144 mm |
| Compression ratio | 10:1 |
| No. of valves per cylinder | 4 |

**Table 2**. In-cylinder pressure sensor characteristics.

| | |
|---|---|
| Pressure range | 0 to 250 Bar |
| Overload | 300 Bar |
| Sensitivity (nom) | -37.0 pC/Bar |
| Natural frequency | >215kHz |

The data are collected by carrying out spark sweep tests on the entire engine operating field. This means that the SA is changed, keeping constant the load, in terms of the trapped air mass, and the speed. This data in this paper is called as “Steady-State Data” and is used for the model development and initial validation.

Figure 1 shows the operating points that are tested for different engine speeds, loads, and $\lambda$ equal to 1, 0.75, and the mapped values, and the quality of fuel used is RON95 (Research Octane Number) gasoline. The values of the engine load and the engine speed have been normalized with respect to their maximum and converted into a percentage, and this criterion of normalization for these indexes is followed in the entire paper for confidentiality reasons. In this work, the engine load is identified with the stoichiometric trapped air mass. This means that when the engine load is mentioned it is referred to as an index comparable with volumetric efficiency.

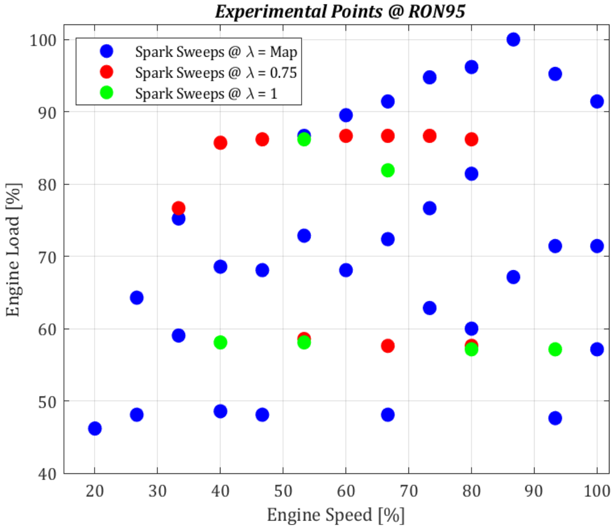

**Figure 1**. Engine operating points tested with different lambda values, using RON95.

For each steady state condition shown in the previous figure, the SA is varied by performing a spark sweep test. The in-cylinder pressure signals are recorded for 200 consecutive cycles, for each value of SA. The value of recorded cycles (200) has been previously used in other works [23] by the authors and, for this reason, it represents the reference for the evaluation of the statistical properties of the combustion and knock indexes in this work. Moreover, the signals coming from all the sensors mounted on the engine are recorded after the steady-state condition is reached and this means that only the cycle-to-cycle variation has to be filtered by averaging the 200 cycles.

The in-cylinder pressure signals are used to calculate the main combustion indexes offline. Importantly, the MFB50 is calculated from the Cumulative Net Heat Release (CNHR) estimated from the low-passed-filtered in-cylinder pressure curves, with a cut-off frequency of 3 kHz. For abnormal combustions, many indexes can be used to estimate the knock intensity. In this work MAPO is used and evaluated from the following equation:

$$MAPO = max\left(\left|p_{filt}\right|_{\theta_0}^{\theta_0+\phi}\right) \tag{1}$$

Where $p_{filt}$ is the windowed and bandpass filtered in-cylinder pressure signal, $\theta_0$ is the crank angle corresponding to the beginning of the window of calculation and ϕ is the width of the window. In this case the cut-off frequencies are 5 and 25 kHz respectively.

During the experimental activity, also some transient tests are carried out and the collected data have been used to develop the innovative training procedure and the accuracy index proposed in this work. Such tests in this paper are called as “Transient Data” and are used to further validate the models and it covers the whole engine operating field. The engine load and speed profiles performed at the bench during these tests are shown in the Figure 2. The values of SA have been normalized and 1 unit equals 10° of Crank Angle (CA) and this criterion for such indexes has been followed in the entire paper.

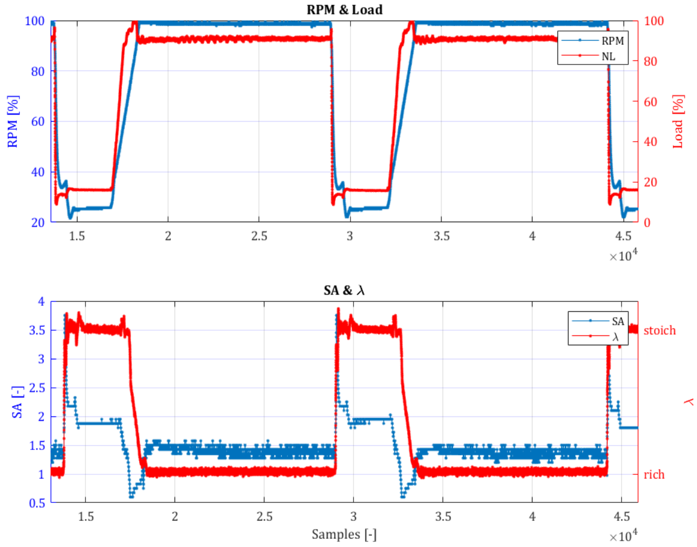

**Figure 2**. Engine load and speed transient profiles.

## Automatic Calibration Algorithm

The automatic calibration procedure of the ANN-based model is explained by considering the example of the MFB50 index, but the same process is implemented to model the other combustion and knock indexes. There are several hyper-parameters to be tuned while calibrating an ANN and the Figure 3 shows the process flow that is followed by the algorithm to find the optimal configuration for the network.

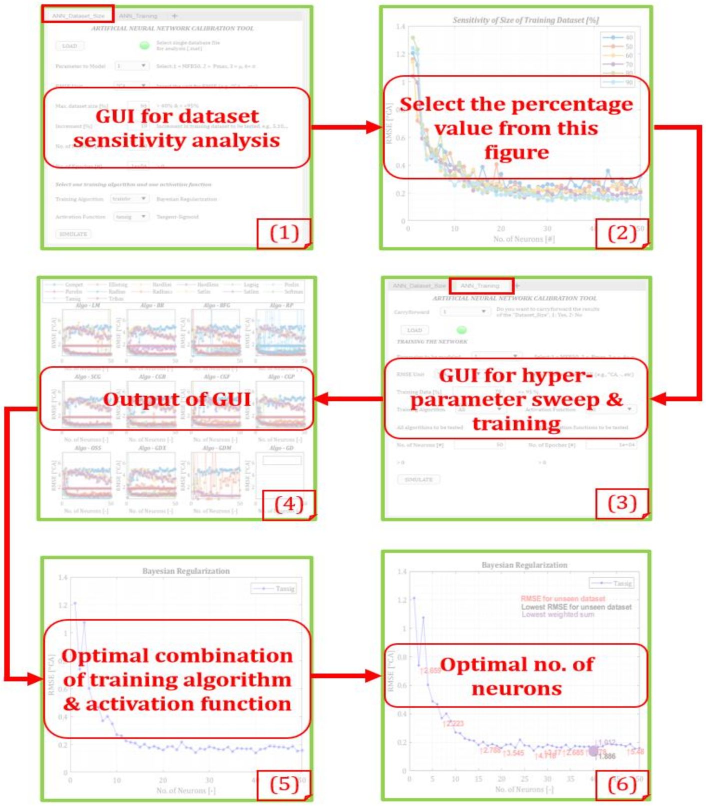
GUI for dataset sensitivity analysis
(1)
Select the percentage value from this figure
(2)
GUI for hyper-parameter sweep & training
(3)
Output of GUI
(4)
Optimal combination of training algorithm & activation function
(5)
Optimal no. of neurons
(6)
ARTIFICIAL NEURAL NETWORK CALIBRATION TOOL
Sensitivity of Size of Training Dataset [%]
No. of Neurons [#]
Bayesian Regularization
Tansig

**Figure 3**. Process flow for automatic calibration.

The data as mentioned in the experimental campaign section, used to develop the models is divided in two main categories which are as follows:

- Steady-state data: This is the dataset that is used to train the models as shown in Figure 1 and it is used for finding the optimal combination of the training algorithm and activation function, explained in the following paragraphs. The data is then divided randomly into three sets, the percentage for each set is selected by the user, which are as follows:
  - Training dataset: A part of the "seen data" is used by the network to learn the phenomenon being modelled.
  - Validation dataset: A part of the "seen data" is used for the validation of the network during training.
  - Test dataset: A part of the "seen data" that the network didn't see during the training is used for evaluating the model that learned the phenomenon using the "training dataset" and the "validation dataset" as mentioned in above.
- Transient data: This is the dataset as shown in Figure 2 which is never seen by the network during training. This is used for further validation of the model and to find the optimal number of neurons for the given model, explained in the following paragraphs.

The optimization algorithm proposed is based on the definition of a novel accuracy index, in this work named "Rank", which evaluates the weighted sum of certain quantities as mentioned in the Equations (2) and (3).

$$Rank_{TAAF} = A * mean(RMSE_{testdata}) + B * std(RMSE_{testdata}) + C * min(RMSE_{testdata}) \tag{2}$$

$$Rank_{NN} = D * (RMSE_{testdata}) + E * (RMSE_{unseendata}) \tag{3}$$

Where:

- Rank$_{TAAF}$ = Rank for selecting the optimum combination of the training algorithm and the activation function
- Rank$_{NN}$ = Rank for selecting the optimum number of neurons
- A, B, C, D, E = weights to be assigned by the user, ranging from 0 to 1. For the MFB50 index, the values of weights used are A, B, C = 0.33, D = 0.6 and E = 0.4
- mean = mean value of the signal
- std = standard deviation of the signal
- min = minimum value of the signal

The developed algorithm is divided into different steps to find the optimal configuration in terms of the percentage of the training dataset, the training algorithm, the activation function, and the number of neurons. The main steps of the algorithm are described in the following paragraphs.

The first window of the GUI is used to evaluate the sensitivity of the results to the percentage of the dataset used to train the network. From the developed GUI, the algorithm selected for the network training is the Bayesian Regularization and the activation function is the Tangent-Sigmoid (tansig). From the literature, it can be seen that this combination is the most commonly used for these kinds of applications [24]. Since the goal is to find the optimal size of the training dataset, and all the optimization algorithms and the activation functions are designed to work at their best performance, any algorithm could be used in the step. During this analysis, the dataset is divided into different values of percentage to be used as the training sets, for example from 40% to 90% and the network is trained for all these values. For each value of the percentage of the training dataset, the number of neurons is varied from 1 to 50, calculating the RMSE and R$^2$ between the experimental and the simulated values referred to in the test dataset.

In the second step of the procedure, the GUI outputs the results shown in the Figure 4. It is the evaluation of RMSE for each value of the percentage of the training dataset as a function of the number of neurons. As seen from the Figure 4, using a low percentage of training data, for example, 40%, could result in overfitting issues of the model, whereas, using a high value, for example, 90%, there are no real benefits in terms of the accuracy of the model. This high value of training dataset size could result in making the model more specific rather than having a good generalization (the issue is well known as the so-called overfitting). Furthermore, this figure also helps to decide the RMSE that can be considered as an admissible threshold for filtering the trends and helps in selecting the weights (in Equations 2 and 3) required in the next steps. A good compromise between the accuracy and the generalization would be to use 70% data as training data in this case.

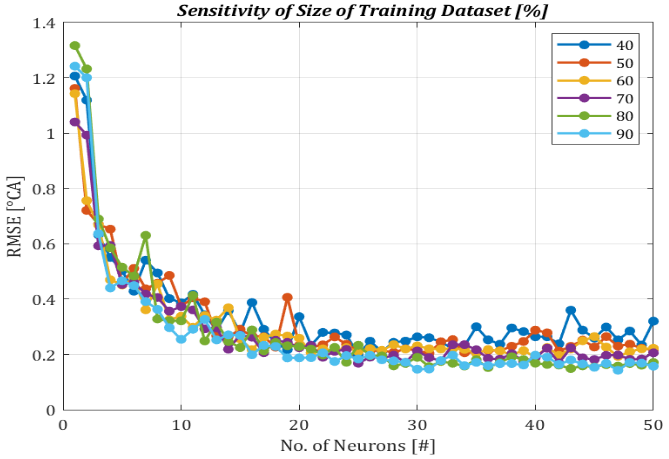

**Figure 4**. Training dataset sensitivity analysis.

Then using the size of the training dataset as 70%, the GUI carries out a sweep test for all the combinations of training algorithms and the activation functions. Each of these combinations is tested for 1 to 50 neurons. The number of epochs is set to 8000 (as mentioned in the introduction section). The list of all the training algorithms and the activation functions available in Matlab [25,26] is shown in Table 3.

**Table 3**. List of all the training algorithms and the activation functions available in Matlab.

| Training Algorithms | |
|---|---|
| **Acronym** | **Full form** |
| LM | Levenberg Marquardt |
| BR | Bayesian Regularization |
| BFG | Broyden Fletcher Goldfarb Shanno Quasi Newton |
| RP | Resilient Backpropogation |
| SCG | Scaled Conjugate Gradient |
| CGB | Conjugate Gradient with Powell/Beale Restarts |
| CGF | Fletcher-Powell Conjugate Gradient |
| CGP | Polak-Ribiére Conjugate Gradient |
| OSS | One Step Secant |
| GDX | Variable Learning Rate Gradient Descent |
| GDM | Gradient Descent with Momentum |
| GD | Gradient Descent |
| **Activation Functions** | |
| **Acronym** | **Full form** |
| compet | Competitive transfer function |
| elliotsig | Elliot sigmoid transfer function |
| hardlim | Positive hard limit transfer function |
| hardlims | Symmetric hard limit transfer function |
| logsig | Logarithmic sigmoid transfer function |
| netniv | Inverse transfer function |
| poslin | Positive linear transfer function |
| purelin | Linear transfer function |
| radbas | Radial basis transfer function |
| radbasn | Radial basis normalized transfer function |
| satlin | Positive saturating linear transfer function |
| satlins | Symmetric saturating linear transfer function |
| softmax | Soft max transfer function |
| tansig | Symmetric sigmoid transfer function |
| tribas | Triangular basis transfer function |

In the end, the output of the previous step is the Figure 5. Only a few cases are reported in the figure to improve the readability, but such results are obtained for all the combinations of the training algorithms and the activation functions mentioned in Table 3. It shows the RMSE comparison for all the combinations of the training algorithms and the activation functions reported on the number of neurons axis. As seen from Figure 5, there are many combinations that have a high value of RMSE with respect to others and can be directly excluded from the analysis thereby saving some unnecessary computational effort. As an example of this, the yellow, blue, and purple curves in the top-left hand graph of Figure 5 can be considered. Figure 6 shows the same analysis but for $R^2$ for the same data. Since the trend of the $R^2$ curves are similar as the ones for RMSE i.e., the lowest RMSE has the highest $R^2$, these curves are not used in the calculations, instead are used only for visualization.

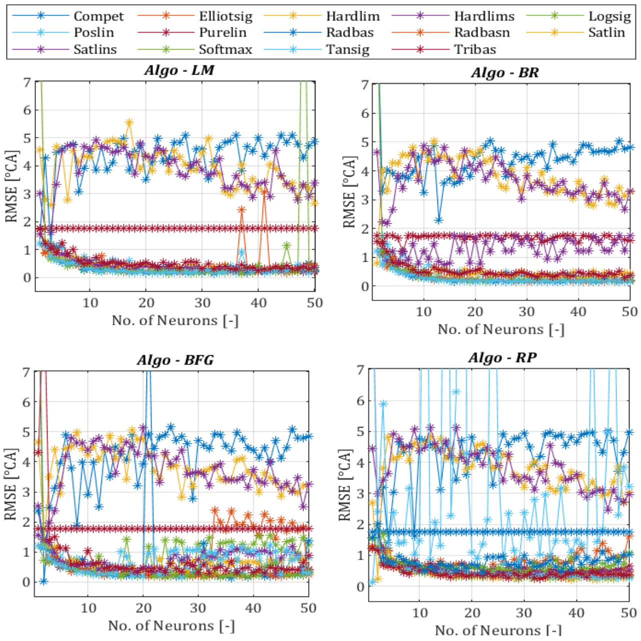

**Figure 5**. RMSE comparison for all the combinations of the training algorithms and the activation functions, as a function of number of neurons. The title on each graph indicates the training algorithm and the legend indicates all the activation functions.

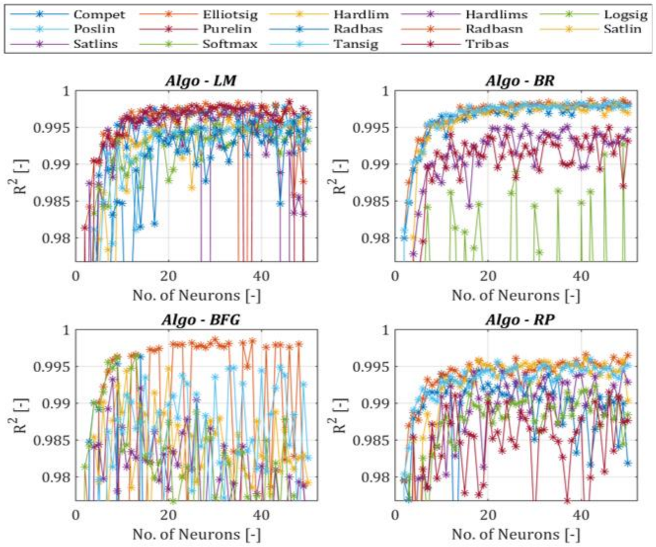

**Figure 6**. R$^2$ comparison for all the combinations of the training algorithms and the activation functions, as a function of number of neurons. The title on each graph indicates the training algorithm and the legend indicates all the activation functions.

To exclude the unwanted signals, the results are filtered based on a threshold set for the mean value of the signal, given by the Equation (4).

$$Filter_{Threshold} = mean(signal) \tag{4}$$

Where, the signal refers to each curve showed in the Figure 6

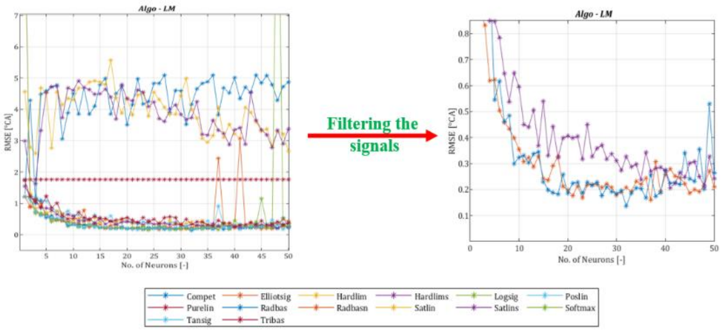

**Figure 7**. Filtering the unwanted signals from Figure 5. This is an example of filtering the signals in the first graph of Figure 5.

For this MFB50 model, as seen in Figure 7 on the left plot, many signals have a high value of RMSE. The Filter$_{Threshold}$ is set as 0.4°CA and following the Equation (4), the signals whose mean value is above this threshold are excluded from further calculations, as seen from the right plot in Figure 7. This procedure is applied to all the graphs in Figure 5.

After the first filtering mentioned above, the "Rank$_{TAAF}$" introduced in the Equation (2) is evaluated. Figure 8 explains why the statistical parameters such as the mean, the standard deviation, and the minimum value, are used for evaluating the rank. As seen in Figure 8, the blue curve (compet) has the lowest minimum value but has an oscillating trend with a high RMSE. The pink curve (purelin) has the lowest standard deviation, but the mean RMSE is high with respect to the others. The green curve (logsig) has instead a smoother trend with respect to other curves. Thus, evaluating the rank based on the statistical parameters ensures avoiding a combination having an inaccurate combination of the hyper-parameters having a local minimum.

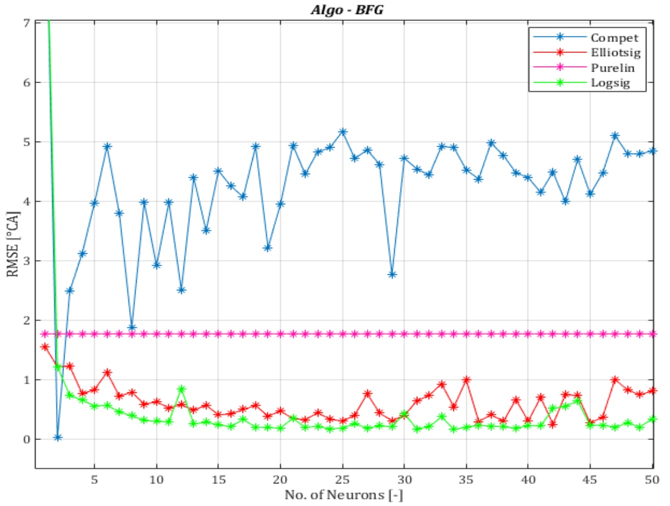

**Figure 8**. Trend of the activation function showing the trend of each of them. Only a few functions are shown in this figure to have a good readability.

The weights in the Equation (2) must be set by the user in the GUI and Figures 6 and 7 help in assigning the weights in the Equation (2). The combination of the training algorithm and the activation function having the lowest "Rank$_{TAAF}$" is selected as the optimal one, as shown in Figure 9. For the considered MFB50 model, the optimal training algorithm is the Bayesian Regularization, and the optimal activation function is the Tangent-Sigmoid (Tansig).

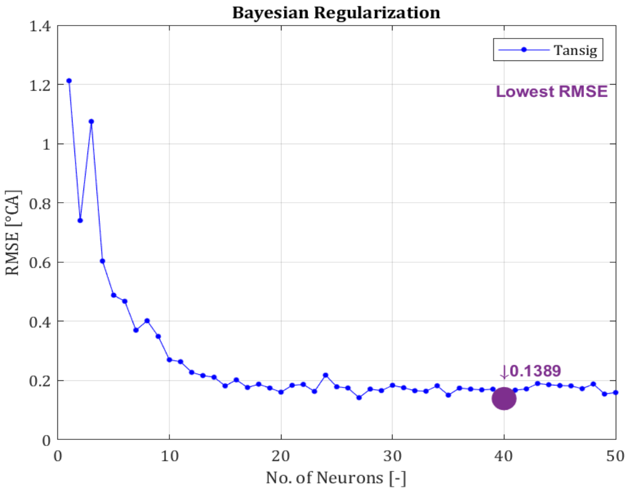

**Figure 9**. Optimal combination of the training algorithm and the activation function.

At this point, there are two possible ways to move forward depending on the availability of a separate dataset in this paper known as "Transient data", namely "Method A" and "Method B", explained as follows:

- Method A: This method is followed when a separate "Transient data" datasetis not available. Following this method, the procedure suggests an optimal number of neurons based on the lowest RMSE achieved in the previous step, for example, neuron number 40 as shown in Figure 8. Alternatively, the user could also select the number of neurons based on his experience which corresponds to an acceptable value for RMSE, for example, neuron number 20 in Figure 9.

- Method B: This method is followed when a separate "Transient data" datasetis available. Again "Rank$_{NN}$" is evaluated following the Equation (3). This weighted sum is calculated between the RMSE of the Steady-State data (test set) and the RMSE of the Transient data, for each neuron. The optimal number of neurons is the one having the lowest "Rank$_{NN}$" as shown in Figure 10. The figure shows the RMSE value for the Transient data (red color) only for a few points to have a good readability of the figure. This rank ensures that the model has high accuracy along with good generalization.

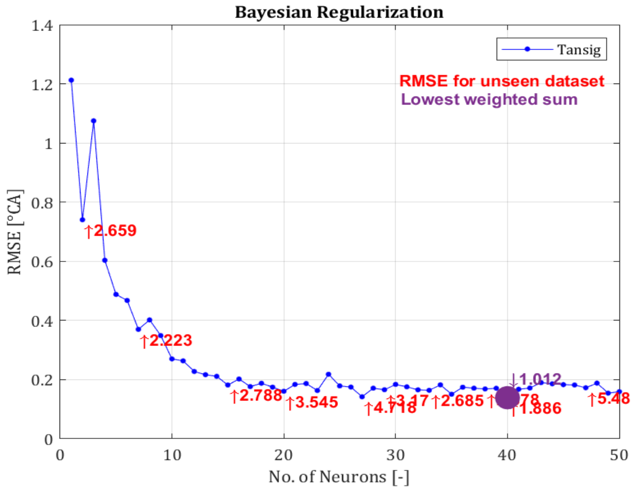

**Figure 10**. Optimal number of neurons selection.

Such functionality represents an extremely important aspect in the network training program. Indeed, one of the main issues related to the automatic, graphical procedures available in commercial programs to train neural networks, is related to the reliability of the prediction on a dataset that is completely different to that used to train the model. Introducing this function and the Rank index in the proposed algorithm allows to significantly increase the robustness of the network. The final output of the proposed procedure contains the network configuration as summarized in Table 4.

**Table 4**. Final ANN configuration for the MFB50 model.

| | |
|---|---|
| Training Algorithm | Bayesian Regularization |
| Activation Function | Tangent Sigmoid |
| Number of neurons | 40 |

The time taken to develop the models largely depends on factors such as the amount of data available, number of epochs used, type of hardware used, etc. For this MFB50 model (following all the steps as shown in Figure 3) the time required was approximately 15 hours. Similar times have been observed for all the other models developed in this work. Table 5 summarizes the hardware used while developing the models.

**Table 5**. Hardware used for developing all the models.

| | |
|---|---|
| Processor | Intel(R) Core (TM) i5-8265U<br>CPU @ 1.60 GHz 1.80 Gz |
| Read Only Memory (RAM) | 8 GB |
| System type | 64-bit operating system |

## Results

This section describes the models that are developed, and the results achieved following the procedure explained in the previous section. The MFB50 and the Pmax model are developed following the "Method B" i.e., availability of the "Transient data", while the IMEP and the knock models are developed following the "Method A" i.e., non-availability of the "Transient data".

### *MFB50 Model*

The inputs to the combustion phase model i.e., MFB50 are the engine speed, load, SA, and $\lambda$. The model is trained to reproduce the mean value between all the cylinders. The final configuration of the ANN is as mentioned in Table 4. Figure 11 shows the results achieved for the testing dataset (from Steady-State data) (left graphs) and the Transient data (right graph). The MFB50 model has an accuracy with an RMSE of 0.139 °CA and R$^2$ of 0.9998 in the steady-state condition. For the transient tests as shown in the Figure 11 (right plot), the simulated results don't follow very well the experimental results (highlighted in the graph by a black ellipse) because the inputs (engine load, speed, SA, and lambda) in this part of the tests (Transient data) were not present in the " Steady-State data" used for training the ANN and they represent the low engine load and speed points in the engine operating range. Furthermore, the results achieved in the rest of the part of the transient test (medium-high engine load and speed) are accurate and they represent more important engine operating points. The Table 6 shows the statistical analysis performed on the error which is evaluated as the difference between the experimental and the simulated results, as shown in the Equation (5)

$$Error = Experimental - Simulated \qquad (5)$$

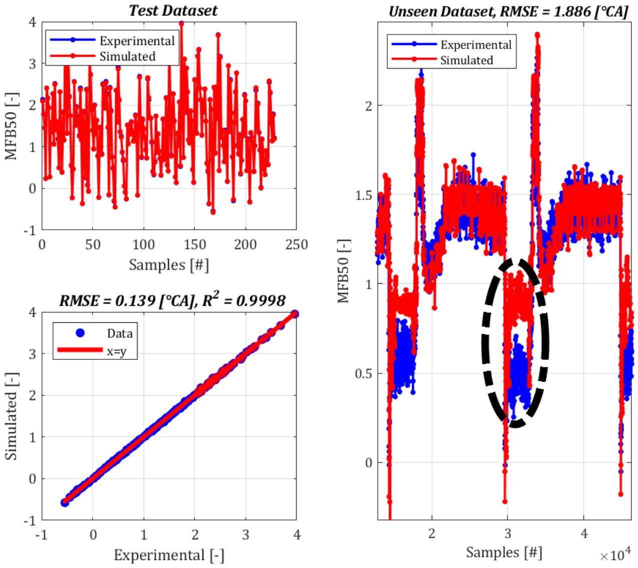

**Figure 11**. Comparison between the experimental and the simulated results for MFB50.

As seen from Table 6, the statistical analysis is carried out for two cases, firstly the entire Transient data named "Full". The second case is referred to the same Transient data but excludes the points that were not a part of the training dataset used to train the network (highlighted in black in the Figure 11). Here the "μ" is the mean value and "σ" is the standard deviation, which are evaluated for the error as defined in Equation (5). As seen from this analysis, excluding the points not present in the training dataset, the model has a better performance, i.e., having a σ of 1.5087 means that approximately 86.6% of the samples are within an error ± 1.5σ or in other words, 86.6% of samples are in within the range almost equal to the RMSE of the model, which is 1.886°CA. The Figure 12 shows the histogram comparison between the two cases, highlighting the results achieved for the statistical analysis.

**Table 6**. Statistical analysis for the Transient data.

| | μ | σ |
|---|---|---|
| $Error_{Full}$ | -0.1938 | 2.1462 |
| $Error_{Partial}$ | 0.2961 | 1.5087 |

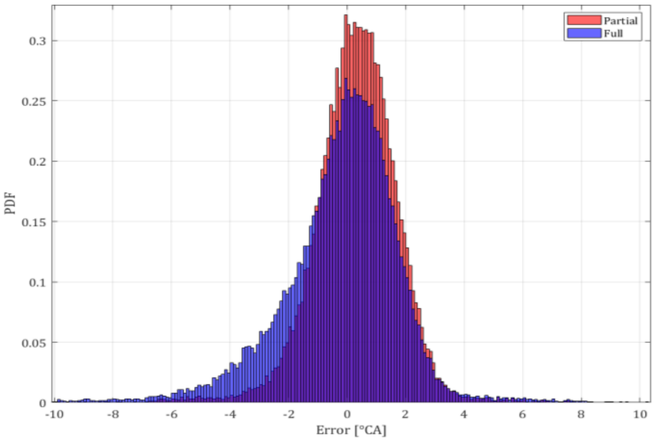

**Figure 12**. Statistical analysis for the error calculated in two cases.

This consideration of evaluating the "Rank" allows to highlight how important is the evaluation of the model performance also on an Transient data during its development and configuration. The definition of a novel accuracy index that weighs the results for two different datasets is strategic to prevent an undesired or unphysical value estimated by the network. The values of MFB50 have been normalized and 1 unit equals 10°CA, and this normalization criterion is followed for this index in the entire paper, while the calculation chain for this model is shown in the Figure 13.

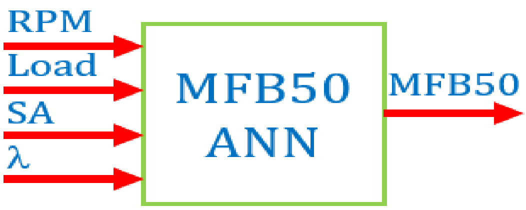

**Figure 13**. Calculation chain for MFB50.

## *Pmax Model*

The inputs to the Pmax model are the engine load and the MFB50. Also, in this case, the model predicts the mean cylindrical value. While developing this model, the MFB50 considered is the experimental one because this model was developed independently of the MFB50 model. The final ANN configuration for this model is mentioned in Table 7. Figure 14 shows the results achieved for the testing dataset from the Steady-State data (left graphs) and the Transient data (right graph). As seen from the results, the model has good accuracy with an RMSE of 1.682 Bar and $R^2$ of 0.9942 in the steady-state condition and RMSE of 5.940 Bar in the transient condition. The model shows very good accuracy also under transient conditions. The part of the results (highlighted by black ellipse) in Figure 14, has a small error when compared to the MFB50 model (as highlighted in Figure 11), because the Pmax is not very sensitive to the MFB50 in the low engine load and speed range. Furthermore, when the models are coupled, it is more important to have high accuracy in the more important operating points, which the model is able to achieve. The values of Pmax have been normalized with respect to the maximum one, and this normalization criterion is followed for this index in the entire paper, while the calculation chain for this model is shown in Figure 15.

**Table 7**. Final ANN configuration for the Pmax model.

| | |
|---|---|
| Training Algorithm | Bayesian Regularization |
| Activation Function | Tangent Sigmoid |
| Number of neurons | 22 |

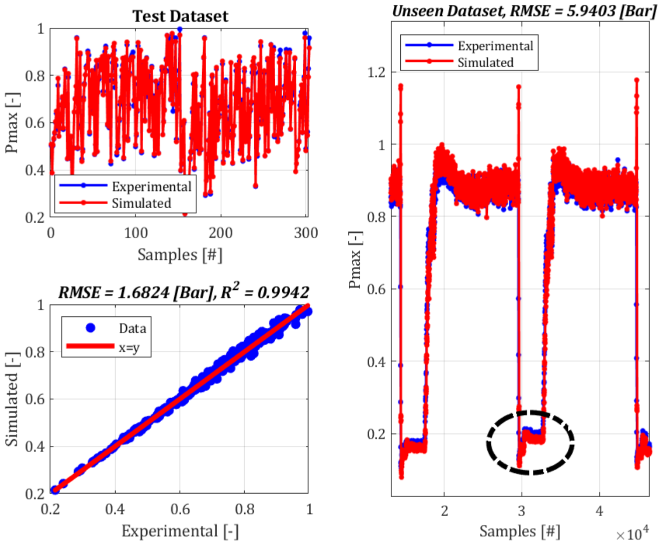

Figure 14. Comparison between the experimental and the simulated results for Pmax.

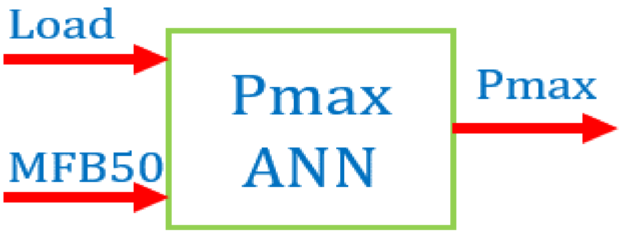

**Figure 15**. Calculation chain for Pmax.

## *IMEP Model*

The IMEP model is approached in a similar way as the Pmax model i.e., the inputs to the model are the engine load and the MFB50. As mentioned earlier, this model is developed using the first approach i.e., "Method A" (non-availability of Transient data). The final ANN configuration of this model is mentioned in Table 8. The values of IMEP have been normalized with respect to the maximum one, and this normalization criterion is followed for this index in the entire paper Figure 16 shows the results achieved for this model, having an RMSE of 0.616 Bar and $R^2$ of 0.9801. The IMEP model has not been validated with an  Transient data for two main reasons. Firstly, to demonstrate that the “Method A” can give accurate results. Furthermore, the IMEP model is approached with a similar concept to the Pmax model, and the results achieved for the Pmax model have good accuracy, hence it can be considered that also the IMEP model will have similar results on the unseen dataset.

**Table 8**. Final ANN configuration for the IMEP model.

| | |
|---|---|
| Training Algorithm | Bayesian Regularization |
| Activation Function | Satlin |
| Number of neurons | 17 |

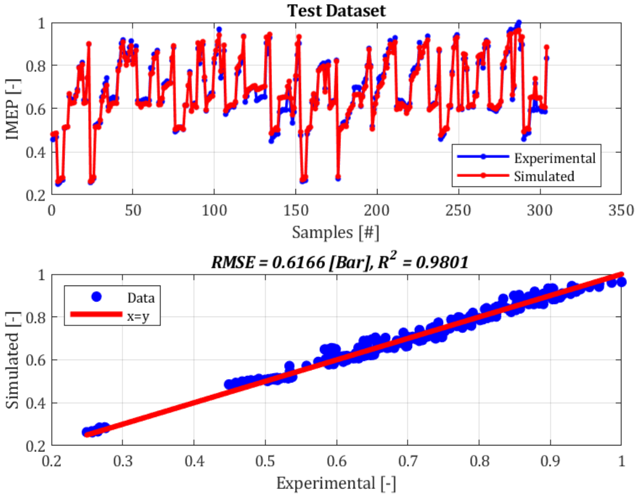

**Figure 16**. Comparison between the experimental and the simulated results for IMEP.

## *Knock Model*

It is well known that knock is a stochastic phenomenon that can be best studied by statistical analysis. For a fixed engine speed, load, SA, and lambda, the cyclic MAPO statistical distribution can be described by a Gamma or a Log-Normal distribution Probability Density Function (PDF). Such observation allows considering a statistical approach also for the knock intensity modelling and in this work the 98th percentile of MAPO (MAPO98) is used as the intensity index. The Log-Normal PDF inherits the main properties of the Gaussian PDF which allows to completely determine the distribution with two parameters, i.e., the mean value μ and the standard deviation σ. In the previous works of authors [22], it has been demonstrated the calculation of these two parameters by the following equations:

$$\mu = log(MAPO50) \tag{6}$$

$$\sigma = \frac{log(MAPO98) - \mu}{2.0057} \tag{7}$$

Where:

- MAPO50 corresponds to the 50th percentile of MAPO
- MAPO98 represents the 98th percentile of MAPO
- μ is the mean value
- σ is the standard deviation

Two separate ANN have been trained to model μ and σ using the data from the spark sweep test, which are calculated following equations (6) and (7) on the recorded data of the spark sweep tests. The inputs to these models are the Pmax, the engine speed and load, and lambda and the outputs are μ and σ respectively. The MAPO98 is then evaluated by inverting the equation (7). The final ANN configurations for these two models are mentioned in Table 9. Figure 17 shows the results achieved for the knock model, and as seen an RMSE of 0.457 Bar and $R^2$ of 0.9908 is achieved. Figure 18 shows the calculation chain for evaluating MAPO98.

**Table 9**. Final ANN configuration for the μ and the σ model.

| | μ | σ |
|---|---|---|
| Training Algorithm | Levenberg Marquardt | Bayesian Regularization |
| Activation Function | Logsig | Logsig |
| Number of neurons | 36 | 28 |

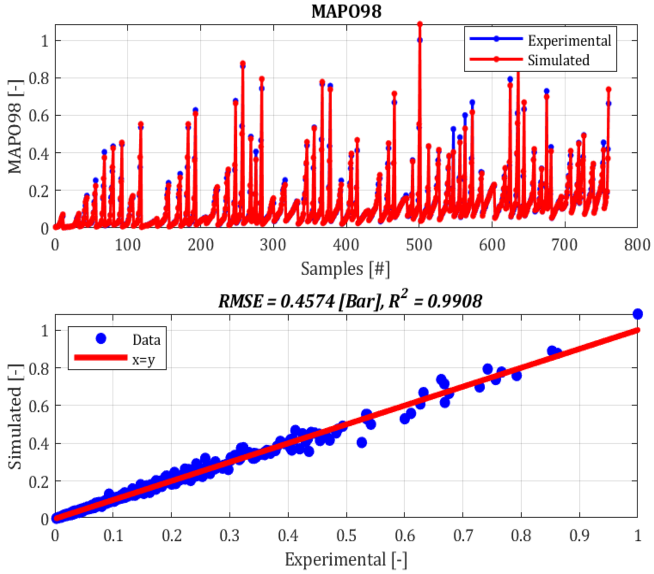

**Figure 17**. Comparison between the experimental and the simulated results for MAPO98.

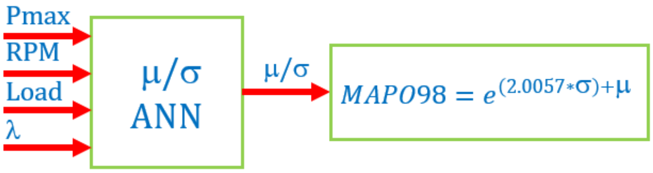

**Figure 18**. Calculation chain for MAPO98.

## Conclusion and Future Development

In the literature, there are no pre-defined procedures available for calibrating the ANN once the modeling issue is well defined and this makes the calibration process an iterative one and a time expensive one.

This paper attempts to address this issue by introducing an automatic calibration algorithm and a novel index to weigh the model accuracy on different datasets, with the final aim to prevent poor performance on a different dataset. The definition of rank ensures the robustness of the model in terms of high accuracy as well as a good generalization in the case where there is an unseen dataset available. In the case where there is no  separate data (called as Transient data in this paper) available, the algorithm is still able to output the configuration of ANN that has high accuracy. As seen from the results achieved, following Method B, the MFB50 model has an RMSE of 0.139°CA and 1.886°CA under the steady-state and the transient condition respectively. The Pmax model has an RMSE of 1.682 Bar and 5.940 Bar under the steady-state and the transient condition respectively. Furthermore, following Method A, the IMEP model has an RMSE of 0.616 Bar while the knock model has an RMSE of 0.457 Bar, in the steady-state condition. From these results, it can be concluded that the ANN configuration given by the algorithm performs well in both the steady-state and the transient conditions and for the different datasets.

A future development of the proposed work will be to include also the automatic selection of the number of hidden layers, which may be required in some cases. Features such as an early stop of training is required to further optimize the training time. It is quite useful to have an input sensitivity analysis feature, which may be required when the number of inputs is large. Although the algorithm is automatic but the weights for determining the rank are to be assigned by the user. Thus, an algorithm will be developed to calculate these weights automatically based on certain criteria/rules. Furthermore, these developed models can be coupled with RT systems and the performance can be evaluated with homologation cycles.

## References

1. AVL Public Discussion, https://www.avl.com/documents/4329920/48266926/AVL+Emission+Test+System+and+Emission+New+Regislation.pdf,17/11/2021.
2. Brusa, A.; Cavina, N.; Rojo, N.; Mecagni, J.; Corti, E.; Ravaglioli, V.; Cucchi, M.; Silvestri, N. Development and Experimental Validation of an Adaptive, Piston-Damage-Based Combustion Control System for SI Engines: Part 1—Evaluating Open-Loop Chain Performance. *Energies* **2021**, *14*, 5367. https://doi.org/10.3390/en14175367
3. Brusa, A.; Cavina, N.; Rojo, N.; Mecagni, J.; Corti, E.; Moro, D.; Cucchi, M.; Silvestri, N. Development and Experimental Validation of an Adaptive, Piston-Damage-Based Combustion Control System for SI Engines: Part 2—Implementation of Adaptive Strategies. *Energies* **2021**, *14*, 5342. https://doi.org/10.3390/en14175342
4. Chiodi, M., Perrone, A., Roberti, P., Bargende, M. et al., "3D-CFD Virtual Engine Test Bench of a 1.6 Liter Turbo-Charged GDI-Race-Engine with Focus on Fuel Injection," *SAE Int. J. Engines* 6(3):1834-1845, 2013, https://doi.org/10.4271/2013-24-0149
5. Kaechele, A., Chiodi, M., and Bargende, M., "Virtual Full Engine Development: 3D-CFD Simulations of Turbocharged Engines under Transient Load Conditions," *SAE Int. J. Engines* 11(6):697-713, 2018, https://doi.org/10.4271/2018-01-0170
6. Northrop, W. and Zarling, D., "One-Dimensional Modeling of a Thermochemical Recuperation Scheme for Improving Spark-Ignition Range Extender Engine Efficiency," *SAE Int. J. Adv. & Curr. Prac. in Mobility* 2(2):543-550, 2020, https://doi.org/10.4271/2019-24-0066
7. De Castro Radicchi, F., Vieira, J.M.G., Fonseca, L., Barros, J.E.M. and Valle, R.M., 2015. *Numerical Analysis of an Internal Combustion Engine Using One-Dimensional Computational Models* (No. 2015-36-0377). SAE Technical Paper, https://doi.org/10.4271/2015-36-0377
8. Scocozza, G., Silvagni, G., Brusa, A., Cavina, N. et al., "Development and Validation of a Virtual Sensor for Estimating the Maximum in-Cylinder Pressure of SI and GCI Engines," SAE Technical Paper 2021-24-0026, 2021, https://doi.org/10.4271/2021-24-0026
9. De Bellis, V., Malfi, E., Bozza, F., KUMAR, D. et al., "Experimental and 0D Numerical Investigation of Ultra-Lean Combustion Concept to Improve the Efficiency of SI Engine," *SAE Int. J. Adv. & Curr. Prac. in Mobility* 3(4):1993-2008, 2021, https://doi.org/10.4271/2021-01-0384.
10. Ranuzzi, F., Cavina, N., Brusa, A., De Cesare, M. et al., "Development and Software in the Loop Validation of a Model-based Water Injection Combustion Controller for a GDI TC Engine," SAE Technical Paper 2019-01-1174, 2019, https://doi.org/10.4271/2019-01-1174
11. Ranuzzi, F., Cavina, N., Scocozza, G., Brusa, A. et al., "Experimental Validation of a Model-Based Water Injection Combustion Control System for On-Board Application," SAE Technical Paper 2019-24-0015, 2019, https://doi.org/10.4271/2019-24-0015
12. Gambarotta, A. and Lucchetti lng, G., "Control-Oriented “Crank-Angle” Based Modeling of Automotive Engines," SAE Technical Paper 2011-24-0144, 2011, https://doi.org/10.4271/2011-24-0144
13. Szybist, J., Wagnon, S., Splitter, D., Pitz, W. et al., "The Reduced Effectiveness of EGR to Mitigate Knock at High Loads in Boosted SI Engines," *SAE Int. J. Engines* 10(5):2305-2318, 2017, https://doi.org/10.4271/2017-24-0061
14. Mecagni, J., Brusa, A., Cavina, N., Corti, E. et al., "Control-Oriented Exhaust Gas Temperature Modelling Based on Wiebe Equation," *SAE Int. J. Engines* 14(5):697-712, 2021, https://doi.org/10.4271/03-14-05-0042
15. Brusa, A., Mecagni, J., Cavina, N., Corti, E. et al., "Development and Experimental Validation of a Control-Oriented Empirical Exhaust Gas Temperature Model," SAE Technical Paper 2020-24-0008, 2020, https://doi.org/10.4271/2020-24-0008
16. Shethia, F., Mecagni, J., Brusa, A., and Cavina, N., "Development and Software-in-the-Loop Validation of an Artificial Neural Network-Based Engine Simulator," SAE Technical Paper 2022-24-0029, 2022, https://doi.org/10.4271/2022-24-0029.
17. MISYRIS, George S.; VENZKE, Andreas; CHATZIVASILEIADIS, Spyros. Physics-informed neural networks for power systems. In: *2020 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 2020. p. 1-5. DOI: 10.1109/PESGM41954.2020.9282004
18. VIANA, Felipe AC, et al. Estimating model inadequacy in ordinary differential equations with physics-informed neural networks. *Computers & Structures*, 2021, 245: 106458. https://doi.org/10.1016/j.compstruc.2020.106458
19. Maldonado, B., Kaul, B., and Szybist, J., "Artificial Neural Networks for In-Cycle Prediction of Knock Events," SAE Technical Paper 2022-01-0478, 2022, https://doi.org/10.4271/2022-01-0478.
20. Fang X, Zhong F, Papaioannou N, Davy MH, Leach FC. Artificial neural network (ANN) assisted prediction of transient NOx emissions from a high-speed direct injection (HSDI) diesel engine. *International Journal of Engine Research*. 2022;23(7):1201-1212. doi:10.1177/14680874211013254
21. Murugesan, S., Srihari, S., and Senthilkumar, D., "Investigation of Usage of Artificial Neural Network Algorithms for Prediction of In-Cylinder Pressure in Direct Injection Engines," SAE Technical Paper 2022-01-5089, 2022, https://doi.org/10.4271/2022-01-5089.

22. Brusa, A., Mecagni, J., Corti, E., and Silvestri, N., "Application of a Neural-Network-Based Algorithm for the Real-Time Correction of the In-Cylinder Pressure Signal Sensed with a Piezoelectric Washer," *SAE Int. J. Engines* 16(5):663-679, 2023, https://doi.org/10.4271/03-16-05-0039.
23. Cavina, N., Brusa, A., Rojo, N. and Corti, E., 2018. "*Statistical Analysis of Knock Intensity Probability Distribution and Development of 0-D Predictive Knock Model for a SI TC Engine* (No. 2018-01-0858)". SAE Technical Paper, https://doi.org/10.4271/2018-01-0858
24. ROBERTSON, Dennis; PRUCKA, Robert. *Neural Network Design of Control-Oriented Autoignition Model for Spark Assisted Compression Ignition Engines*. 2021. DOI: https://doi.org/10.4271/2021-24-0030.
25. Matlab Documentation, The Math Works, 2021, https://it.mathworks.com/help/deeplearning/ug/train-and-apply-multilayer-neural-networks.html
26. Matlab Documentation, The Math Works, 2021, https://it.mathworks.com/help/deeplearning/ug/choose-a-multilayer-neural-network-training-function.html;jsessionid=281eabc2c22a6da1f137b7337085

## Abbreviations

| | |
|---|---|
| 0D | Zero Dimensional |
| 1D | One Dimensional |
| 3D | Three Dimensional |
| ANN | Artificial Neural Network |
| CA | Crank Angle |
| CFD | Computational Fluid Dynamics |
| CNHR | Cumulative Net Heat Release |
| FNN | Feedforward Neural Network |
| GUI | Graphical User Interface |
| IMEP | Indicated Mean Effective Pressure |
| MAPO50 | 50th Percentile of Maximum Amplitude of Pressure Oscillation |
| MAPO98 | 98th Percentile of Maximum Amplitude of Pressure Oscillation |
| MFB50 | 50% Mass Fraction Burned |
| PDF | Probability Density Function |
| Pmax | Maximum In-cylinder Pressure |
| $R^2$ | R-Squared |
| RMSE | Root Mean Squared Error |
| RON | Research Octane Number |
| RT | Real Time |
| SA | Spark Advance |