Application of Machine Learning Methods for the Development of Internal Combustion Engines – An Overview

Youssef Beltaifa, Shahida Faisal, Maurice Kettner

Karlsruhe University of Applied Sciences
Gas Engine Laboratory (GenLab)

Youssef.beltaifa@h-ka.de
shahida7085@gmail.com
Maurice.kettner@h-ka.de

Abstract. Machine Learning (ML) has a strong potential to improve the performance and effectiveness of several technologies and processes. In recent years, ML has gained in importance, primarily due to its matchless success in image recognition and computer games. These ML accomplishments have motivated to transfer and adapt its algorithms and modeling methods to most scientific disciplines. For instance, in mechanical engineering, ML is coming to hold a crucial position ranging from value chain optimization (production) to substitution of complex simulation models (research and development). In the case of traditional research and development approach, the analysis and optimization of a process are implemented according to the understanding of the governing mechanisms described by physical and mathematical rules. On the contrary, the intelligence of the ML method originates from the extraction of trends and laws based on data patterns, which produces surprisingly good results in many cases. However, it is not entirely evident why it performs so well. One of the most challenging mechanical engineering topics is the improvement of the Internal Combustion Engine (ICE) towards higher efficiency and lower negative impact on the environment. ICEs are very complex systems, which involve high-speed reciprocating motions, transient gas flow and combustion chemistry. Thus, the application of ML methods for ICEs opens new perspectives regarding the modelling, control and maintenance. These topics are addressed in detail in the course of this paper, based on the most relevant published results found in the literature, to provide an overview to the actual research and development of ICE using ML methods.

Keywords: Machine Learning; Mechanical Engineering; Internal Combustion Engines; Modelling; Control; Predictive Maintenance

1 Introduction

Internal combustion engines will maintain their position as major power source during the coming decades, particularly for heavy-duty applications [1, 2]. Future internal combustion engines have to comply with tightening legislative emission-limits, high fuel-energy conversion-efficiency, affordable prices and customer requirements. To reach this target, engine researchers worldwide are working on innovative exhaust aftertreatment systems, alternative combustion processes, bio- and renewable fuels, lightweight materials, modern lubricants and advanced manufacturing processes. Within the development of innovative combustion processes (research focus of Gas Engine Laboratory at Karlsruhe University of Applied Sciences) mainly experimental (mostly at the engine test bench) and numerical investigations (0D, 1D and 3D-CFD) are performed. Engine tests are expensive (costly metrology, etc.) and very time-consuming. Moreover, numerical simulations are very dependent on the validation of the implemented physics-based models and necessitate in many cases a large computational capacity. Considering this facts, different alternative approaches that enable saving costs, time and computational power are required. One of the possible solutions that has been increasingly used in recent years is machine learning.

Machine learning is a branch of knowledge dealing with training computers to forecast output values or to classify things without having been explicitly programmed for such function. Machine learning success in many areas like image/speech recognition, effective internet search, self-driving cars is mainly lead by the availability of huge datasets. Machine learning methods can be categorized into two main groups: supervised and unsupervised algorithms, as shown in Figure 1, which depicts some of the most used machine learning algorithms.

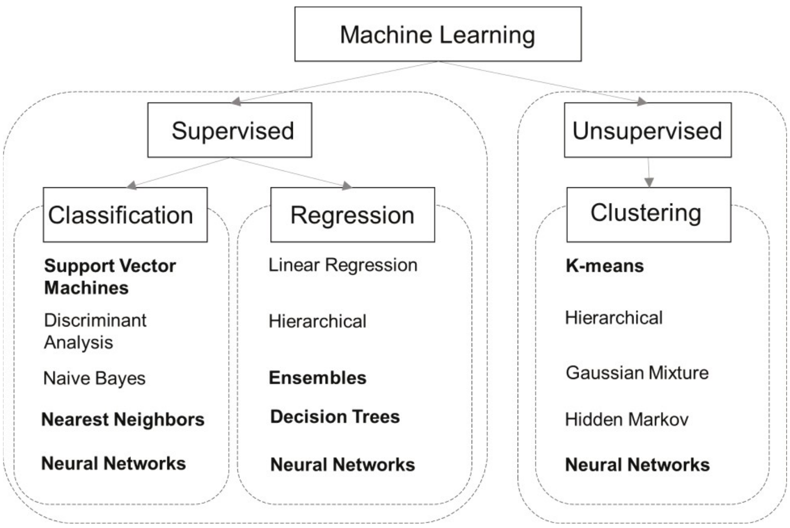

Figure 1. Classification of the most common machine learning algorithms

The methods marked in bold in Figure 1 are the methods that have been used the most in recent studies dealing with internal combustion engines, which are collected and analyzed within this paper. These works are classified within this paper depending on the intended use of machine learning into three categories: Prediction of engine operation parameters and emissions, anomaly detection and predictive maintenance, and real-time engine control. These three topics are covered throughout this paper in detail.

2 Prediction of Engine Operation Parameters and Emissions

Many studies [3-18] have demonstrated that engine combustion associated parameters and emissions can be predicted accurately using neural networks over a wide range of operating conditions, given that the training data provides good knowledge of the system's behavior. The combination of fast-computational time and the network's ability to analyze broad non-linear problems can potentially replace expensive exhaust gas sensors (FID, Gas Chromatograph, etc.) and physics-based, computationally intensive engine modeling approaches. Multi-Layer Perceptron (MLP) is a conventional artificial neutral network (ANN) structure that is commonly used for the prediction of engine operating parameters and exhaust gas components. MLP consists of input, hidden and output layers, as seen in Figure 2 on the left.

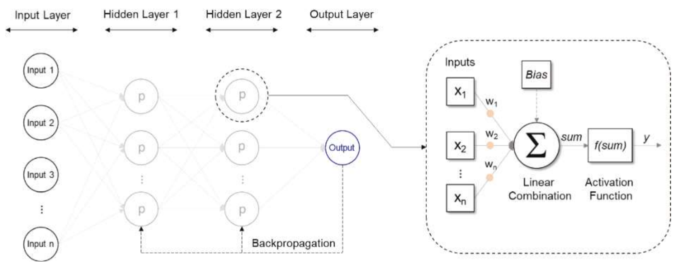

Figure 2. Structure of MLP with two (as example) hidden layers (left), Structure of Perceptron (right)

As shown in Figure 2 on the right, each input is assigned to a weighting factor, representing the importance of the input factor given by the model when predicting the output. Activation functions are employed in the hidden and output neurons, allowing mapping the non-linear relationships between outputs and inputs. For the model training, MLP uses among others gradient descent backpropagation algorithm, where the goal is to minimize the modeling error, meanwhile the weights between neurons are gradually adjusted. Usually, the network training takes place using the Levenberg-Marquardt (LM) backpropagation algorithm, known for high computational efficiency. LM algorithm is a curve-fitting method for solving nonlinear least-squares problems. LM combines the two minimization algorithms gradient descent and Gaussian-Newton to minimize the sum of the squared errors between the fitted model function and the experimental data [19]. MLP with the weighting approach can also be used to have an insight on the dependency of the model output on the input parameters. As an example, the authors in [20] analyzed the relative importance of the in-cylinder parameters affecting the NO_x and HC model output by extracting the saved weights from the trained network. The results yielded that both HC and NO_x were commonly dependent on the engine load and IMEP. The model showed a significant dependency of the NO_x emissions on the peak pressure in the combustion chamber, which is physically reasonable. Higher peak pressures in the combustion chamber are associated with high charge temperatures, which result in turn in a high temperature oxidation of the diatomic nitrogen in the combustion air and the formation of "thermal" NO_x. Further studies demonstrating the success of artificial neural networks in predicting and modeling of engine-operation associated parameters and emissions are summarized in Table 1. For these studies, the statistical efficiency of the models lies between 94% and 99.9%. It is important to notice, that MLP with backpropagation is the most frequent encountered machine learning approach in the field of the research and development of internal combustion engines.

Table 1. Summary of MLP applications for the prediction of engine operation characteristics and emissions found in literature

Author	Output Parameters	Inputs Parameters
Danaiah et al. [3]	Brake specific fuel consumption CO, CO_2, HC, NO_x, O_2	Tert butyl alcohol blend percentage Load Engine speed
Uslu et al. [4]	Brake specific fuel consumption Brake mean effective pressure HC, CO, NO_x	Speed Fuel blend (i-amyl alcohool in gasoline) Compression ratio
Gürgen et al. [5]	Cyclic variability	Fuel mixtures Engine speed
Togun et al. [6]	Specific fuel consumption Torque	Iginition timing Throttle angle Engine speed
Tasdemir et al. [7]	Specific fuel consumption Torque Power HC	Intake valve advancement speed
Roy et al. [8]	Specific fuel consumption NO_x, CO_2	Load Diesel injected Fuel injection pressure EGR
Maurya et al. [9]	Ringing intensity at different conditions of a hydrogen HCCI engine	Combustion duration Combustion phasing Equivalence ratio Engine speed Inlet valve temperature
Martinez et al. [10]	NO_x	Injection timing Torque Intake pressure Engine speed Ignition timing Throttle angle
Hariharan et al. [11]	Brake specific fuel consumption NO_x, HC, CO	Load H_2 massflow
Mehra et al. [12]	Torque Brake specific fuel consumption NO_x, CO, HC, CH_4 (methane slip)	Engine load Excess air ratio Spark timing
Ghobadian et al. [13]	Torque Specific fuel consumption CO, HC	Blend percentage (Waste cooking oil) Engine speed
Kapusuz et al. [14]	Brake specific fuel consumption	Power Torque Fuel mass flow
Aydin et al. [15]	Brake specific fuel consumption NO_x, HC, CO	Fuel injection pressure Biodiesel blend Load
Akkouche et al. [16]	Airflow Pilot fuel flow Exhaust temperature	Biogas mass flow (CNG engine) Methane contents Power
Oguz et al. [17]	Torque Power Fuel mass flow Brake specific fuel consumption	Engine speed Fuel type
Cay et al. [18]	Specific fuel consumption Air-fuel ratio CO, HC	Blend percentage (Methanol / gasoline) Engine speed Torque

Further neural network concepts/architectures have been used in other studies. Taghavi et al. [21] considered in addition to the MLP network the non-linear autoregressive network with exogenous inputs (NARX) as well as the radial basis function (RBF) network for the prediction of start of combustion (SOC) of a HCCI engine. Input parameters were the intake mixture characteristics (Air-Fuel-Ratio, EGR, intake mixture temperature) as well as the engine speed. The NARX algorithm has, depending on the usage (training or prediction) two structures: The series-parallel (or open loop) and the parallel (or closed loop) architectures. The two network architectures are shown schematically in Figure 3.

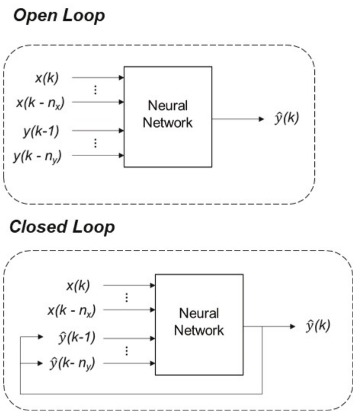

Figure 3. NARX networks architectures

The series-parallel architecture is used for training: the prediction at time-step $t+1$ is provided based on real input and output values at the current time step t, as well as those from the previous n time steps, as shown in Eq. 1. The pure feedforward architecture of the series-parallel (open-loop) structure is applied during training due to the fast static backpropagation [22]. By providing the real input-output pairs during training, the model is able to make future prediction with excellent accuracy. After training, the model produces a final set of adjusted weights, which minimizes the error between the predicted and the true output values. The adjusted weights together with the activation functions approximate the nonlinear mapping function F in Eq. 1. During the prediction stage, the open-loop structure is converted to a closed-loop architecture. Instead of using the real output when making future prediction (time step t+1), the trained model takes the output predicted by itself from the current time step as input, as well as those from the previously n time-steps, as shown in Eq. 2.

$$\hat{y}(t+1) = F\begin{pmatrix} \boldsymbol{y(t)}, y(t-1), \dots, y(t-n_y), x(t+1), \\ x(t), x(t-1), \dots, x(t-n_x) \end{pmatrix} \tag{1}$$

$$\hat{y}(t+1) = F\begin{pmatrix} \boldsymbol{\hat{y}(t)}, \hat{y}(t-1), \dots, \hat{y}(t-n_y), x(t+1), \\ x(t), x(t-1), \dots, x(t-n_x) \end{pmatrix} \tag{2}$$

Additionally, Taghavi et al. [21] applied the radial basis function (RBF) networks also for predicting the SOC using the same input parameters as in the case of the NARX network. The RBF network typically uses only an input layer, a single hidden layer and an output layer [23], as shown in Figure 4 on the left.

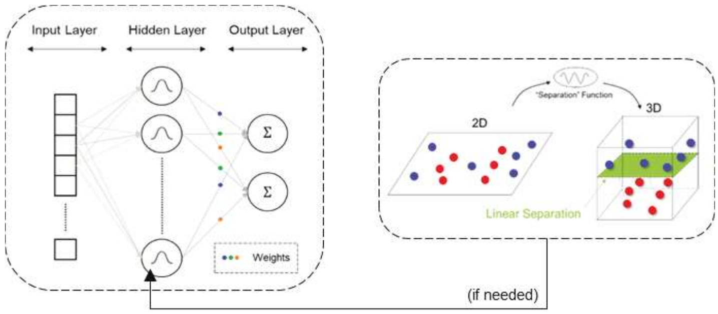

Figure 4. RBF network architecture (left), Schematic description of data dimensionality increase enabling linear separation (from 2D to 3D) (right)

The RBF network is shallow and its behavior is strongly influenced by the nature of the special hidden layer, which performs a computation based on a comparison with a prototype vector [23]. The structure and computations performed in the hidden layer are the key to the power of the RBF network. Here, a hybrid calculation involving two stages takes place. Within the first stage, the linear separability should be ensured: If needed, a projection of the original data points into a higher dimensionality, so that they become linearly separable, is performed. This is based on the Cover's theorem on separability of patterns [24]. For a simplified understanding, Figure 4 on the right shows this step schematically. The second stage is the RBF (Radial Basis Functions) computation, which is based on the comparison of the input units $\bar{X}$ with the prototype vectors $\bar{\mu}_i$ in the hidden layer units according to the equation (3) [23].

$$h_i = \varphi_i(\bar{X}) = \exp\left(-\frac{\|\bar{X} - \bar{\mu}_i\|^2}{2 \cdot \sigma_i^{\,2}}\right) \tag{3}$$
$$i \in \{1, \dots, \mathrm{m}\}$$

m is the total number of the hidden units. Each of these m units is created to have a high impact on a particular cluster of points, which is closest to its prototype vector $\bar{\mu}_i$ [23]. Therefore, m can be regarded as the number of clusters used for modeling, and it represents an important hyper-parameter available to the algorithm [23]. Each unit has a bandwidth σ_i, which is often the same for all units with the different prototype vectors [23]. After the RBF calculation in the hidden layer, the outputs from the RBFs are weighted and summed by a simple connection to the output layer. The values of the weights need to be learned in a supervised way, dealing with the specific studied case [23]. On the contrary, the hidden layer is trained in an unsupervised way [25]. This involves several parameters such as the prototype vectors, the bandwidths and the number of hidden neurons m. Elaborate description about the determination methods of these parameters can be found in [23]. In comparison to MLP and RBF, the NARX network featured a better prediction accuracy, reaching R = 0.99933 [21].

Another machine learning process used for the prediction of engine-operation related parameters is the Ensemble modeling. For the prediction of the performance as well as efficiency of an engine converted from the diesel CI to the natural gas SI combustion process, Liu et al. [26] applied ensemble methods (bagging and boosting) and compared their prediction performances. The model output was the indicated mean effective pressure (IMEP). Input parameters were spark timing, fuel/air-ratio and engine speed with overall 153 sets of data (122 for training and 31 for testing). "*Unity is strength*": This statement describes in three words the core idea behind the strength of ensemble methods in machine learning. Such methods improve the predictive performance of a single model by training multiple models and combining their predictions [27]. The base models building the ensemble model are "weak" learners, which feature either a high bias or much variance. These are combined within the ensemble method in such a way that they build a strong learner. The combination strategy of the base learners enables to group the ensemble methods in two main categories, depending on how the base learners are generated [28]. The first category is "bagging". Here the individual learners are created independently and their generation can be parallelized [28]. The second category, called "boosting", creates individual learners sequentially in a very adaptive way [28]. Both ensemble methods are shown schematically in Figure 5. For the first step of the bagging algorithm, multiple bootstrap samples (data subsets) are created. These subsets are almost independent datasets created from the original one using random selection [26]. It is important to notice, that the size of the original dataset should be large enough compared to the size of the bootstrap samples so that they are "sufficiently" independent. Subsequently, one "weak" learner (usually the same) is

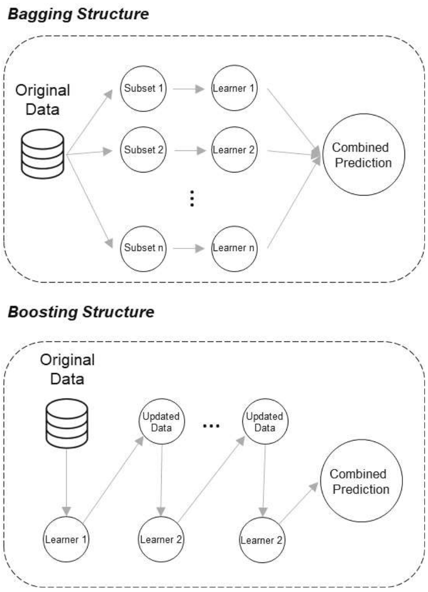

Figure 5. General structure of bagging (top) and boosting (bottom) ensemble algorithms

fitted for each bootstrap sample. The predictions of the base learners are then combined to the final prediction of the ensemble model in some kind of weighting process [27]. The combination of the base learners within the bagging method enables the reduction of the variance compared to the variance levels of the single base learners [28]. Therefore, base models with low bias but high variance are more suitable for bagging. Concerning the boosting algorithm, it is not suitable for parallelized computation. Boosting starts with training a first base "weak" learner and then adapt the distribution of the training data according to the output of the base learner such that incorrectly classified samples will have increased consideration from subsequent basic learners [28]. In other words, each new base learner focuses on the most difficult samples (wrongly predicted by the previous learner), so that we get a strong ensemble model with low bias. Hence, base learners with low variance but high bias are suitable to be combined within boosting ensemble methods. Liu et al. [26] found that boosting outperformed bagging, can deal with data set with uneven distributed conditions among the operating range, and provided a high accuracy prediction ($R^2 = 0.9623$) even for low frequency cases, which are poorly presented in the original data set.

3 Anomaly Detection and Predictive Maintenance

With recent developments, powertrain systems are becoming more complex. Understanding this complexity and dealing with associated particular problems/failures requires evolved methods. New detection methodologies involving machine learning and predictive diagnostics have become the need of the hour [29]. In this frame, Farsodia et al. [30] proposed an approach combining unsupervised learning and clustering to detect anomalies, which may occur in engines or after-treatment-systems (ATS). To validate their strategy, Farsodia et al. [30] addressed the example of the backpressure problem occurring in the diesel particulate filter (DPF) of an automotive diesel engine. Figure 6 depicts schematically the approach proposed by Farsodia et al. [30].

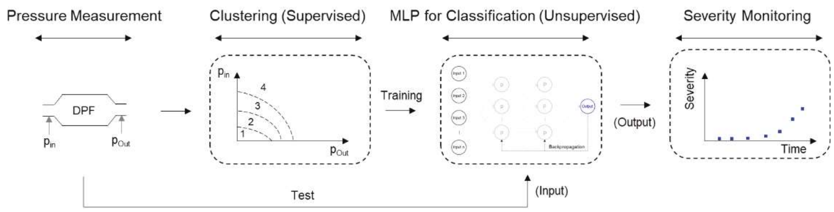

Figure 6. Schematic description of the approach proposed by Farsodia et al. [30] for anomaly detection

As shown in Figure 6, the pressure values before and after the DPF are measured and clustered in a supervised way using the k-means clustering algorithm. Here, the data set will be distributed into "k" clusters. Each cluster has a centroid, which is defined by averaging (taking mean) of the assigned data. First, centroids are determined randomly. Then, data points from the dataset are arranged to the nearest available centroid. The positions of the centroids change within an optimization process until further movement of centroid is not possible. The algorithm is one of the most commonly used techniques for clustering purposes, as it quickly finds the centers of the clusters. Detailed specifications to the k-means clustering algorithm can be found in [31]. In a further step, a classification MLP is trained with the defined classes (clusters) from the supervised clustering step and the associated data. The trained classification MLP is then used in a third step to predict in an unsupervised manner the operating mode of the DPF. For a better monitoring of malfunction cases, Farsodia et al. [28] defined a "severity factor", which enables a time dependent tracking of the DPF functionality degradation. The "severity factor" is derived based on the relative density of data (e.g. malfunction, backpressure, etc.) with respect to total available data points [30]. This "severity factor" gives a pre-warning about any component's malfunction, which will enable the end user to take necessary preventive measures [30].

In a further case, Farsodia et al. [30] presented a methodology involving the weighted k-nearest neighbor (w-kNN) algorithm to predict the temperature shoot-up events in a DPF, which are harmful for the ATS from thermal aging and safety perspectives. kNN is among the simplest statistical learning tools in density estimation, classification as well as regression and known to be trivial to train and easy to code [32]. The difference between the standard and the weighted kNN is that in the weighted approach the prediction of a test point is more depended on the nearest observations [30]. In other words, the k points within the neighborhood of the test point do not contribute equally to the final decision of the test point. Indeed, the closer an observation is from a test point, the more it contributes to its classification. For deeper insight into the w-kNN-methodology, please refer to [33]. After defining the most probably governing parameters on the temperature shoot-up event (engine speed, torque, airflow, HC injection quantity, etc.), Farsodia et al. [30] classified the training dataset, containing temperature shoot-up events, into three different risk categories: "high", "medium" and "low", using w-kNN within a supervised learning process. Category "high" risk implied that there are very high chances that there will be temperature shoot-up post DOC. When testing the trained model with test data from the same vehicle, the model released a warning signal about 60 seconds before the temperature shoot-up event occurred. The algorithm derived from the data is "smart" enough to detect the difference between the high-end temperatures and shoot-up events. However, the excellent beforehand prediction performance was not precisely explained by the authors. Especially, the relationship between the algorithm behind the occurrence of the warning signal and the previous classification step was not discussed.

For a 2.4L diesel excavator engine, Jang et al. [34] proposed also an anomaly detection model, which is depicted schematically in Figure 7. The main idea of the proposed approach is to extract abundant features from gathered data using an autoencoder and then to distinguish between normal and abnormal operating conditions with help of a one-class support vector machine (OCSVM). First, data was collected from 123 different sensors at high frequency (one value every 0.1 s) over 12 days. Due to the large learning dimension, raw collected data cannot be applied to the autoencoder. Therefore, the authors used statistical values instead (median, variance, deciles, etc.). This enabled the reduction of the data amount and the expression of data characteristics more prominently. In a second data-dimensionality-reduction step, the autoencoder is applied to the derived statistical indicators. Autoencoders are neural networks that can automatically (unsupervised) learn useful features from data [35]. Autoencoders work by compressing the data into a latent-space representation also known as bottleneck, and then reconstructing the output from this representation. Jang et al. [34] used compressed features from the latent space of the autoencoder network as input for the classification algorithm, which is the OCSVM, which is used in the context of pattern classification to discriminate between two classes [36]. More details to support vector machines can be found in [37]. Ten days of "healthy" measurement data were used to train the OCSVM model. The anomaly classification performance was evaluated using data from two days, where faulty events were present. The model accuracy reached up 73%. However, the model achieved an excellent recall score with 83%, indicating the model reliability to ignore false alarms.

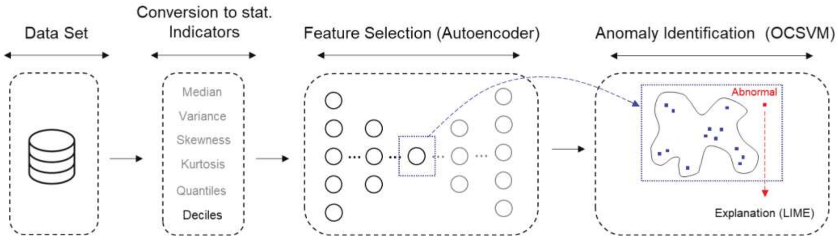

Figure 7. Schematic description of the approach proposed by Jang et al. [34] for anomaly detection

In a classification task, the positive or negative results are insufficient without explaining the classifier's decision-making. Therefore, Jang et al. [34] used Local Interpretable Model-agnostic Explanation (LIME) to get more in-depth interpretation regarding the most critical factors contributing to the classification results. LIME is an algorithm for providing interpretable explanations for the non-interpretable (black box) ML models such as neural networks. An in-depth explanation to the LIME approach can be found in [38].

4 Real-Time Engine Control

Conventionally, ICEs control is based on map-calibrations tuned by full factorial or design of experiments processes. To reach engine efficiency targets, manufacturers are increasing the number of actuators [39], leading to an increase in the calibration design space and thus affecting the real-time capability of the control unit, especially for transient operating conditions. Thus, new control techniques, which can better deal with increasing actuators number, are developed. In this context, Egan et al. [40] introduced a hybrid modelling approach involving the non-linear model predictive control (nMPC) in combination with static and dynamic (time-dependent behavior) artificial neural networks. nMPC is an advanced control strategy that has the greatest acceptance in the industry, because it provides an intuitive approach to the optimal control of systems subject to constraints [41]. Nevertheless, it has its drawbacks, mainly the large amount of calculation required, since an optimization problem is being solved at every sampling time [41]. Thus, non-linear MPC use for ICE is usually limited due to the short period available between engine cycles (~25ms at 5000 rpm) and the limited computational power of automotive control units [42]. Alone the evaluation of non-linear engine-models and its linearization take about 60%-75% of the total computational time per nMPC iteration [41]. Taking into consideration that neural networks can computationally efficient capture non-linear behavior and have the ability to be linearized in minimal time [40], Egan et al. [40] proposed to replace traditional engine modeling methods by artificial neural networks and use them within the nMPC framework, as shown schematically in Figure 8. Their aim was to accelerate the nMPC processing time and thus facilitating its integration into the engine control unit. Egan et al. [40] found that the proposed control system successfully controls the investigated engine with tractable computational load, opening doors for the application of their approach for future Engine Control Units.

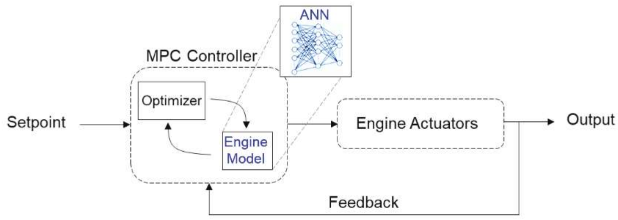

Figure 8. MPC architecture with ANN as engine modeling approach, approach proposed by Egan et al. [40]

5 Summary and Outlook

This paper deals with existing applications of machine learning in the field of internal combustion engine development. Most of the work found in literature handles empirical model building with the use of artificial neural networks, especially the MLP structure, which is quite suitable for mapping non-linear processes occurring in an IC engine. Other structures for modeling and predicting engine-related parameters have also been found in the literature, such as the NARX and the RBF networks. Further studies considered ensemble methods, which are well suitable for modeling engine parameters (especially the boosting algorithm).

In addition to model building, machine-learning methods in the field of combustion engines are also used for the predictive maintenance and anomaly detection. Especially, Clustering (k-means clustering, k-nearest neighbors, support vector machine, etc.) and “deep learning” (autoencoder, convolutional neural network, etc.) methods are used for these purposes. In this process, XAI (explainable artificial intelligence) methods (e.g. Local Interpretable Model-agnostic Explanation) were also employed to get more in-depth explanations and interpretations for the machine learning model decisions. These methods then ultimately allow a more advanced understanding of the engine's behavior.

One more field of machine learning application in the engine development segment is the real-time engine control. New engine Control systems face the challenge of dealing with growing engine complexity and thus increasing computational intensity. In this context, artificial neural networks offer the possibility to reduce the computational effort without affecting the number of actions to be managed with a given time slot. Indeed, the MLP structure have the advantage that it can simply map the engine operation (highly non-linear) and be easily linearized (from non-linear to linear), which motivates for its integration into existing control systems involving computing intensive optimizers.

However, machine learning also has its drawbacks. One cannot expect any magic from machine learning algorithms. Indeed, simple learning programs are unable to learn complex concepts from few input data. To deal with this fact, more data and “smarter” algorithms are needed. Therefore, researchers are increasing the application of deep learning (DL) methods such as convolutional neural networks (CNN) [43], generative adversarial networks (GAN), and autoencoders (AE), which has proven to enable the automatic detection of most significant features during the training phase and to exceed the prediction accuracy of the simpler ML models with conventional human-aided feature extraction. Therefore, we expect an increasing use of deep learning algorithms within the future research and development of internal combustion engines.

References

1. Zhao, L., Ameen, M., Pei, Y., Zhang, Y., Kumar, P., Tzanetakis, T. and Traver, M., "Numerical evaluation of gasoline compression ignition at cold conditions in a heavy-duty diesel engine", SAE Technical Paper, No. 2020-01-0778, 2020
2. Xu, Z., Ji, F., Ding, S., Zhao, Y., Wang, Y., Zhang, Q., Du, F. and Zhou, Y., „Simulation and experimental investigation of swirl-loop scavenging in two-stroke diesel engine with two poppet valves", International Journal of Engine Research, 1468087420916083, 2020.
3. Danaiah P., Kumar P., Rao Y., "Performance and emission prediction of a tert butyl alcohol gasoline blended spark ignition engine using artificial neural networks". Int J Ambient Energy 36:37–41, https://doi.org/10.1080/01430750.2013. 820147, 2013
4. Uslu S., Celik M., "Performance and exhaust emission prediction of a SI engine fueled with I-amyl alcohol gasoline blends: an ANN coupled RSM based optimization". Fuel 265:116922, https://doi.org/10.1016/j.fuel.2019.116922, 2020
5. Gürgen S., Ünver B., Altın İ. "Prediction of cyclic variability in a diesel engine fueled with n-butanol and diesel fuel blends using artificial neural network". Renew Energy 117:538–544, https://doi.org/10.1016/j.renene.2017.10.101, 2018
6. Kara Togun N., Baysec S. "Prediction of torque and specifc fuel consumption of a gasoline engine by using artificial neural networks". Appl Energy 87:349–355, https://doi.org/10.1016/j. apenergy.2009.08.016, 2010
7. Tasdemir S., Saritas I., Ciniviz M., Allahverdi N. "Artificial neural network and fuzzy expert system comparison for prediction of performance and emission parameters on a gasoline engine". Expert Syst Appl 38:13912–13923, https://doi.org/10. 1016/j.eswa.2011.04.198, 2011
8. Roy S., Banerjee R., Bose P., "Performance and exhaust emissions prediction of a CRDI assisted single cylinder diesel engine coupled with EGR using artifcial neural network". Appl Energy 119:330–340, 2014, https://doi.org/10.1016/j.apenergy. 01.044, 2014
9. Maurya R., Saxena M., "Characterization of ringing intensity in a hydrogen-fueled HCCI engine". Int J Hydrogen Energy 43:9423–9437, https://doi.org/10.1016/j.ijhydene.2018. 03.194, 2018
10. Martínez-Morales J., Quej-Cosgaya H., Lagunas-Jiménez J. et al., "Design optimization of multilayer perceptron neural network by ant colony optimization applied to engine emissions data". Sci China Technol Sci 62:1055–1064, https://doi.org/10. 1007/s11431-017-9235-y, 2019
11. Hariharan N., Senthil V., Krishnamoorthi M., Karthic S. "Application of artifcial neural network and response surface methodology for predicting and optimizing dual-fuel CI engine characteristics using hydrogen and bio fuel with water injection", Fuel 270:117576, https://doi.org/10.1016/j.fuel.2020.117576, 2020
12. Mehra R., Duan H., Luo S. et al., "Experimental and artificial neural network (ANN) study of hydrogen enriched compressed natural gas (HCNG) engine under various ignition timings and excess air ratios". Appl Energy 228:736–754, https://doi. org/10.1016/j.apenergy.2018.06.085, 2018
13. Ghobadian B., Rahimi H., Nikbakht A. et al. „Diesel engine performance and exhaust emission analysis using waste cooking biodiesel fuel with an artificial neural network". Renew Energy 34:976–982, https://doi.org/10.1016/j.renene.2008.08.008, 2009

14. Kapusuz M., Ozcan H., Ahmad J., “Research of performance on a spark ignition engine fueled by alcohol e gasoline blends using artificial neural networks”. Appl Therm Eng 91:525–534, https://doi.org/10.1016/j.applthermaleng.2015.08.058, 2015
15. Aydin M., Uslu S., Bahattin Çelik M., “Performance and emission prediction of a compression ignition engine fueled with biodiesel-diesel blends: a combined application of ANN and RSM based optimization”, Fuel. https://doi.org/10.1016/j.fuel. 2020.117472, 2020
16. Akkouche N., Loubar K., Nepveu F. et al., ,”Micro-combined heat and power using dual fuel engine and biogas from discontinuous anaerobic digestion”. Energy Convers Manag 205:112407, https://doi.org/10.1016/j.enconman.2019.112407, 2020
17. Oguz H., Sarıtas I., Baydan H., “Prediction of diesel engine performance using biofuels with artificial neural network”. Expert Syst Appl 37:6579–6586, https://doi.org/10.1016/j.eswa.2010. 02.128, 2010
18. Cay Y., Korkmaz I., Cicek A., Kara F., “Prediction of engine performance and exhaust emissions for gasoline and methanol using artificial neural network”. Energy 50:177–186, https://doi. org/10.1016/j.energy.2012.10.052, 2013
19. Henri P. Gavin, “The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems”, Department of Civil and Environmental Engineering, 2020
20. Janakiraman V., Suryanarayanan, S., Saravanan, S., and Rao, G., "Analysis of the Effect of In-cylinder Parameters on NO_x and HC Emissions of a CI Engine Using Artificial Neural Networks," SAE Technical Paper 2006-01-3313, 2006
21. Taghavi, M.; Gharehghani, A.; Nejad, F. Bakhtiari; Mirsalim, M., “Developing a model to predict the start of combustion in HCCI engine using ANN-GA approach”. In Energy Conversion and Management 195, pp. 57–69, doi: 10.1016/j.enconman.2019.05.01, 2020
22. Huo F., Poo A., “Non-linear autoregressive network with exogenous inputs based contour error reduction in CNC machines”. In International Journal of Machine Tools and Manufacture 67, pp. 45–52, doi: 10.1016/j.ijmachtools.2012.12.007, 2013
23. C. C. Aggarwal, Neural Networks and Deep Learning, Springer, 2018
24. F. Samuelson and D. G. Brown, "Application of Cover's theorem to the evaluation of the performance of CI observers," The 2011 International Joint Conference on Neural Networks, pp. 1020-1026, doi: 10.1109/IJCNN.2011.6033334.P, 2011
25. Faris H., Aljarah I., Mirjalili S., Chapter 28 - Evolving Radial Basis Function Networks Using Moth–Flame Optimizer, Handbook of Neural Computation, Academic Press, ISBN 9780128113189, 2017
26. Liu, J, Ulishney, C, & Dumitrescu, CE. "Improving Machine Learning Model Performance in Predicting the Indicated Mean Effective Pressure of a Natural Gas Engine." Proceedings of the ASME 2020 Internal Combustion Engine Division Fall Technical Conference, 2020
27. Sagi O., Rokach L., “Ensemble learning: A survey”, WIREs Data Mining and Knowledge Discovery, Volume 8, Issue 4, 2018
28. Zhou Z., “Machine Learning”, Springer, Nanjing, Jiangsu, China, ISBN 978-981-15-1967-3 (eBook), 2021
29. Stephen M., Machine Learning, Second Edition, 2015
30. Farsodia, M., Pandey, S., and Ganguly, G., “Advance Data Analytics Methodologies to Solve Diesel Engine Exhaust Aftertreatment System Challenges,” SAE Technical Paper 2019-01-5035, doi:10.4271/2019-01-5035, 2019
31. Aristidis Likas, Nikos Vlassis, Jakob J. Verbeek, The global k-means clustering algorithm, Pattern Recognition, Volume 36, Issue 2, Pages 451-461, ISSN 0031-3203, 2003

32. K. S. Ni and T. Q. Nguyen, "An Adaptable k-Nearest Neighbors Algorithm for MMSE Image Interpolation," in IEEE Transactions on Image Processing, vol. 18, no. 9, pp. 1976-1987, doi: 10.1109/TIP.2009.2023706, 2009
33. M. Bicego and M. Loog, "Weighted K-Nearest Neighbor revisited," 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 1642-1647, doi: 10.1109/ICPR.2016.7899872, 2016
34. Jang G-b, Cho S-B. Anomaly Detection of 2.4L Diesel Engine Using One-Class SVM with Variational Autoencoder, ANNUAL CONFERENCE OF THE PROGNOSTICS AND HEALTH MANAGEMENT SOCIETY, 2019.
35. Walter Hugo Lopez Pinaya, Sandra Vieira, Rafael Garcia-Dias, Andrea Mechelli, Chapter 11 - Autoencoders, Machine Learning, Academic Press, Pages 193-208, ISBN 9780128157398, 2020
36. G. D. Fraser, A. D. C. Chan, J. R. Green and D. T. MacIsaac, "Automated Biosignal Quality Analysis for Electromyography Using a One-Class Support Vector Machine," in IEEE Transactions on Instrumentation and Measurement, vol. 63, no. 12, pp. 2919-2930, doi: 10.1109/TIM.2014.2317296, 2014
37. Suthaharan S., "Support Vector Machine. In: Machine Learning Models and Algorithms for Big Data Classification". Integrated Series in Information Systems, vol 36. Springer, Boston, MA, https://doi.org/10.1007/978-1-4899-7641-3_9, , 2016
38. Peltola T., Local Interpretable Model-agnostic Explanations of Bayesian Predictive Models via Kullback-Leibler Projections, Machine Learning, Cornell University, 2019
39. Atkinson, C., "Fuel Efficiency Optimization Using Rapid Transient Engine Calibration," SAE Technical Paper 2014- 01-2359, doi: 10.4271/2014-01-2359, 2014
40. Egan, D., Koli, R., Zhu, Q., and Prucka, R., "Use of Machine Learning for Real-Time Non-Linear Model Predictive Engine Control," SAE Technical Paper 2019-01-1289, doi:10.4271/2019-01-1289, 2019
41. Bordons C., Garcia-Torres F., Ridao M.A. "Model Predictive Control Fundamentals. In: Model Predictive Control of Microgrids". Advances in Industrial Control. Springer, Cham, https://doi.org/10.1007/978-3-030-24570-2_2, 2020
42. Zhu, Q., Prucka, R., Prucka, M., and Dourra, H., "A Nonlinear Model Predictive Control Strategy with a Disturbance Observer for Spark Ignition Engines with External EGR," SAE Int. J. Commer. Veh. 10(1):360-372, doi:10.4271/2017-01-0608, 2017
43. Gofran T., Neugebauer P., Schramm D., „Feature extraction from raw vibration signal and classification of bearing faults using convolutional neural networks", Artificial Intelligence from research to application, The Upper-Rhine Artificial Intelligence Symposium UR-AI, 2019