Cost Efficient Training Method for Artificial Neural Networks based on Engine Measurements

Márton Virt, Máté Zöldy

Department of Automotive Technologies
Budapest University of Technology and Economics
Stoczek u. 6, H-1111 Budapest, Hungary
e-mail: virt.marton@edu.bme.hu, zoldy.mate@kjk.bme.hu

The artificial intelligence is an accurate predictive tool for different kinds of internal combustion engine (ICE) applications. However, the training process can be expensive due to the high computational and measurement costs. This work aims to describe a general methodology that can be applied to cost-efficiently train multilayer perceptron type artificial neural networks with measurement data from ICEs. The created methodology is based on analyses of a high-resolution dataset measured on a commercial diesel engine. Different methods and recommendations are presented for the model creation, evaluation, training method selection, input feature selection and architecture selection. In addition, a method is described in order to select the appropriate measurement resolution that provides proper information for training with minimal fuel consumption. The investigation showed that the presented workflow can reduce calculation time and fuel consumption, while maintaining good model accuracy. The method can be applied for any ICE related artificial neural network problems, but it can also be an aide for other research fields.

Keywords: Artificial Neural Networks; Internal Combustion Engines; Methodology; Cost Reducing

1 Introduction

The ambitious goals of the Paris Agreement has a major effect on the climate policy of the European Union (EU). In order to keep the global warming well below 2°C, the EU is devoted to achieve climate neutrality by 2050 [1]. The Fit for 55 package is a set of proposals that is dedicated to cut greenhouse-gas emissions until 2030 by at least 55% compared to the 1990 level [2, 3, 4]. To meet these climate targets, tremendous efforts and investments are required in many sectors, including the transport sector. The first provisional agreement of the Fit for 55 package was made in 2022. This indicates carbon neutrality for the new

passenger cars and light commercial vehicles by 2035. The original proposal considers tailpipe CO_2 emissions only, thus this can be considered a ban for internal combustion engines (ICE) for these vehicle types [5]. This proposal set off many arguments, since a more holistic view is required to properly define carbon neutrality [6]. Besides the local CO_2 emission, the CO_2 emission of the full lifecycle and the Well-to-Wheel (WTW) CO_2 emission also has to be considered, which means that the internal combustion engines operated with e-fuels can also be climate neutral. Currently the political situation changes dynamically. At this point, the EU decided to permit the registration of new ICE cars after 2035 if they are operated with e-fuels.

From a technical point of view, the electrification is the best solution for applications, where the local pollutant emission has to be minimal. Therefore, the rapid electrification is a good solution for most of the passenger car use cases [7]. Since people have most of their personal experience with passenger cars, the simplest political message for campaigning is to provoke electrification in all segments of transportation. From engineering perspective this is an inadequate solution to achieve climate neutral transportation because there are many segments where the current maturity of the battery technology is insufficient [8]. Among other technologies, the sustainable advanced fuels are crucial to achieve true carbon neutral mobility. For applications, where the mass and volume of the energy storage system is critical, the low energy density of the batteries is problematic. This means that e-fuels can be the best short and medium-term solution for aviation, heavy-duty vehicles and road public transport. For applications where high amount of stored energy is required, such as sailing, the high demand for lithium and other expensive rare materials also limits the usability of batteries. The full elimination of ICE from passenger cars will also be a slow process, especially if the entire world's vehicle fleet is considered. The advanced fuels can help to achieve climate neutrality of passenger cars in this transition period, and they can also reduce the pollutant emission by realizing cleaner combustion.

The aspects described above support the urgent need to develop cheap and sustainable advanced fuels [9]. However, it is challenging to creating new advanced fuel compositions [10, 11]. Usually, time-consuming and complex simulations are necessary to model the fuels' effect on the combustion and emission. The applied tools are expensive and deep knowledge – therefore expensive labor is necessary to build and use the simulation models [9]. The artificial intelligence (AI) could provide an alternative approach to efficiently develop new climate neutral fuels. The AI does not require any knowledge on the real physical or chemical processes due to its empirical nature. Once a representative training dataset is available with the proper input and output features, the AI can learn the processes. Then, it can accurately predict the behavior of the investigated system.

The artificial neural networks (ANN) are among the most common AI methods. The structure of this algorithm is similar to the human brain. The inputs of the biological neurons are the dendrites and the output of them is the axon. Once the sum of the incoming electric signals from other neurons reach a certain threshold, the neuron discharges through its axon towards other neurons. The behavior of a neuron can be modelled with a weighted adder. Each connection between artificial neurons has a weight, and a neuron calculates the weighted sum of the incoming signals. Then, the output of a neuron will be modified by an activation function, and the next neuron will get this activation as its input. ANNs are trained with back propagation (BP) algorithms. These are iterative methods that modify the weights (and biases) of the network in order to reduce the error between the desired and actual output values. The most conventional ANN type for ICE related tasks is the multilayer perceptron (MLP) type neural network. This network has an input layer with the defined input features, and an output layer with neurons that calculate the final values of the output features. Between the input and output layer there is at least one hidden layer with a certain number of neurons. Every neuron of a layer is connected to all of the neurons of the next layer. An MLP network with at least one hidden layer is a universal approximator, thus it can be applied to nonlinear problems as well.

Many articles demonstrated the high accuracy of MLP networks for ICE parameter predictions. [12] successfully used an ANN model to accurately estimate soot concentrations in laminar diffusion flames. Authors of [13] trained a MLP network with steady-state NO_x emission measurements and then they validated it with transient data. The created model performed extremely well on the steady-state dataset with a correlation coefficient (R) above 0.99. The overall R values of the network with the transient NO_x data were 0.93 and 0.88 in two different operating points, which is a moderate accuracy. However, it is still a good result considering that the transient data lies far from the range of the steady state measurement points. In [14] ANN models were used to predict the ignition delay (ID) while different n-heptane, iso-octane and toluene mixtures were used. The ID could be predicted from the ambient conditions and the molar fraction of the components with a correlation coefficient above 0.99. The authors could also accurately predict the research octane number and motor octane number of the mixtures.

The development of a predictive tool that is able to estimate different engine parameters accurately in a wide range of operating points with different fuels requires a proper training dataset. However, the creation of high-resolution datasets is costly, especially with special fuels and expensive compounds. The ANN creation method also have to be effective in order to increase accuracy and reduce training time and computing costs. This means that a proper methodology is required to reduce costs and ensure high accuracy. As a first step of the methodology creation, a high-resolution dataset is needed that can be used in different analyses and optimization processes. In our previous study [15] we

created such a dataset with 6277 samples that covered most of the engine's useful operating range. We were able to accurately predict 10 different emission and combustion related parameters from the engine speed, torque and exhaust gas recirculation (EGR) valve position. Now this high-resolution dataset can be used to establish a methodology for cost efficient training of ANNs with ICE measurement data.

This paper presents a workflow that can be applied for the ANN model creation process of ICE related problems. Several investigation is presented to select good methods for the different ANN creation steps. A measurement grid resolution selection technique is also described to minimize fuel costs of measurements while maintaining good model accuracy. Section 3.2 – 3.6 presents the investigations related to efficient ANN creation. Section 3.7 describes the measurement resolution selection method. The full workflow is summarized in the conclusion section.

2 Experimental Apparatus and Methods

2.1 Measurement System and Dataset Creation

The high-resolution dataset was measured on a Cummins ISBe 170 30 turbocharged, medium-duty commercial diesel engine. This article builds on our previous work [15], therefore, the precise description of the measurement system, the calculation methods, and the process of high-resolution dataset creation can be found there. This section only describes methods related to the current study.

2.2 General Methodology of the Investigations

The aim is to create a methodology that provides accurate models fast. The speed of the methods is described by the calculation times. The algorithms run in standardized conditions, where the only load of the computer is the investigated algorithm, thus the calculation times can be compared. To simplify the investigation, the model accuracy is only described with the determination coefficient (R2) of the models since this is one of the most illustrative indicators. A train (70%), a validation (20%) and a test (10%) dataset is used for the analyses. Usually, the validation R2 is used to describe the model performance. However, there are some analyses where the validation data also influence the created ANN models. An example for this is the architecture selection, where the final topology of the network is selected by investigating the validation R2. In these cases, the validation R2 cannot be used for the final evaluation of the performance, so the test R2 is reported.

Most of the investigations use 3 ANN models that were created in our previous study [15]. The Indicated Mean Effective Pressure (IMEP) model had the highest accuracy in that work, so this is used as the representation of excellent models. The Particulate Matter (PM) emission had an average performance, so this represents models with average accuracy, and the Ignition Delay (ID) model was highly inaccurate, so this demonstrates models with bad performance. Therefore, the evaluation of the methods become more realistic. Most investigations use the same topology and input features that were used in [15]. If there is a difference, it is described in the given section.

3 Establishing Best Practices to Efficiently Train ANNs with ICE Measurement Data

3.1 Identifying Parameters of Interest for Optimizations

Every problem that can be solved with an MLP network is different in many aspects, and there is a lot of ways to achieve a satisfying solution to each problem. The experience of the AI researchers shows that there is no general best practice for ANN model creation and the best possible solution can never be discovered. However, many good solutions can be found and researchers can create guidelines for specific problems to identify these easier. The goal of the presented work is to establish such a guideline to reduce computing efforts and measurement cost in case of ANN developments that focus on a wide ICE operation range.

The output of a MLP network depends on many parameters, such as the architecture and the used calculation methods. The necessary input features have to be selected to have the proper information to accurately calculate the output features. Then, this information has to go through the network that need to have the proper number of hidden layers and number of neurons inside the layers. The neurons need to have an appropriate activation function and initializing methods. Then, the created network has to be trained with one of the many existing training methods. Once the network is trained, its performance is needed to be evaluated with data that was not present in the training process. Although this is not a trivial task since the calculation depends on random factors as well.

In the following subsections, some general good practices will be presented on the selection of activation functions and initializing methods. Then, the 2 most common evaluation method will be analyzed. After this, the performance of 6 commonly used training algorithm will be compared. Thenceforth, the input feature selection and architecture selection methods will be discussed. Finally, a new method is described to identify the necessary resolution of a measurement

grid to achieve good accuracy with minimal fuel consumption during the measurement. The workflow is summarized in the conclusions section.

3.2 Activation Functions and Initialization

The activation function is a vital element of neural networks since it brings nonlinearity in the equations. Without this nonlinearity, a single layer could represent the whole network, and the representation of more complex functions would not be possible. Originally the sigmoid function was the most common activation function for hidden layers. Later the tangent hyperbolic activation function also become common, as it provided easier training and better predictive ability. However, both of these functions have a problem: the saturation. When the weighted sum is too high, the gradient of these activation functions converge to zero, which leads to the so-called vanishing gradients problem. Nowadays, the rectified linear unit (ReLU) is the state-of-art activation function. The ReLU is a piecewise linear function that gives a constant gradient for positive weighted sums. This activation function solves the vanishing gradient problem, thus its usage in the hidden layers is a good practice for MLP networks. For the output layer, a pure linear activation function has to be used since the current mathematical problem is a regression. [16]

Besides the activation functions, the initial value of the network's weight also needs to be determined in order to start the calculations. The weights are initialized to small random numbers, and the initialization method depends on the used activation functions. For sigmoid, tangent hyperbolic and pure linear activation functions usually Xavier initialization is used [17]. This method is not ideal for ReLU activation function, so He initialization is used for that [18]. The networks not only have weights, but biases as well. These also have to be initialized, but usually it is a good practice to set their initial values to zero.

The training of the ANN can start after the initializations. During this process, a loss function that represent the network's error is minimized. For regression problems the mean squared error (MSE), mean squared logarithmic error (MSLE) and the mean absolute error (MAE) can be used as a loss function. The MSLE is usually used when the output values can be really high, and the MAE is usually used when extreme outliers are expected. Generally, the use of MSE is the best practice for scaled data, therefore, this is used in our methodology.

3.3 Performance Evaluation

The performance of the ANNs have to be evaluated in order to demonstrate their predictive ability. The output features have a random nature due to the applied random factors during calculation. This randomness has to be treated for proper and consistent results. The 2 most common methods for this are the repeated

evaluation and the k-fold cross-validation. The repeatability of the results and the evaluation time highly depends on the used techniques. This step affects all later analyses of this work; hence the performance evaluation method is studied first.

During repeated training, the network is trained with the training dataset and evaluated with the validation dataset for multiple times. Then, the reported results are the average of each training-evaluation pairs. Here, the parameter of the method is the number of evaluations (n_{eval}). The other method is the k-fold cross-validation. This technique divides the full dataset into k folds. The network is trained and evaluated for k times, and a different fold is chosen as a validation hold-out dataset for each training and evaluation pairs. Thus the network is always validated with a different fold, and the remaining k-1 folds are used to train the network. The final result is the average of the k evaluation. The parameter of this method is the k number of folds. This section compares these methods with different settings.

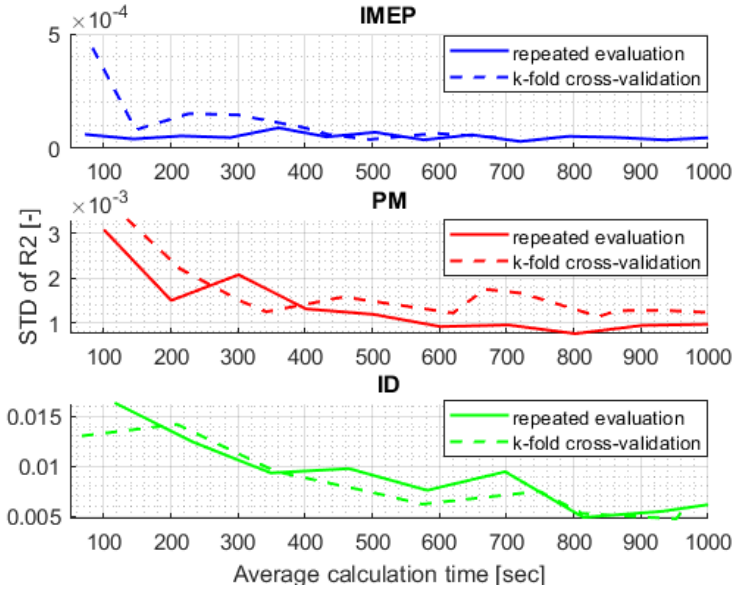

Figure 1
Standard deviations of R2's with repeated evaluation and k-fold cross-validation

The standard deviation (STD) of the R2 shows a decreasing tendency for increasing n_{eval} and k values, but the evaluation time increases. The goal is to identify the method that can achieve smaller STD (thus better repeatability) within the same calculation time. This analysis investigates the n_{eval} between 1 and 20 and the k between 2 and 10. The evaluation with each n_{eval} and k value is repeated 15 times to get the STD of the validation R2 and the mean evaluation time of them. The investigation is done with the IMEP, PM and ID models to investigate the effect of different accuracies.

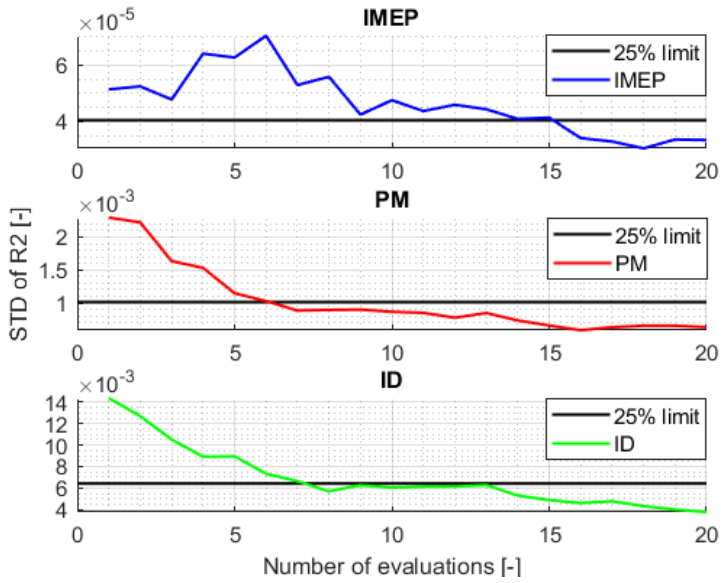

Figure 2
Comparing the standard deviations of R2's with the 25% limit

The results are presented on Figure 1. In case of the IMEP model, the repeated evaluation shows a nearly constant STD as a function of calculation time. The k-fold cross validation has a much higher STD until 400 seconds, then it has nearly the same STD. Therefore. in case of accurate models, the repeated evaluation is better, however. the magnitude of STD is really small for both methods. On the PM model, it is discernible that the STD with the k-fold cross-validation is higher for most of the calculation times. This means that the repeated evaluation is also the better method in case of average accuracy models, however, the difference is not marginable. For the ID model, the two method has a similar performance. The cross-validation becomes slightly better between 400 and 750 seconds. Overall, there is not much difference between the two method's performance at any model accuracy levels, although the repeated evaluation appears to be slightly better. The implementation of this method is also simpler, so this is selected for the workflow.

Next, the optimal number of evaluations have to be determined. On Figure 2, the STD of R2's are presented as a function of n_{eval}. A moving average was applied to the curves with a window of 3 to smooth their characteristics. The value of STD is accepted after it reaches the 25% of the difference between the maximal and minimal STD value. The STD reaches this limit after 15 evaluation repeat for the IMEP model, 6 in case of the PM model, and 7 in case of the ID model. The STD is really low in case of the highly accurate models, thus the 15 repeating is unnecessary. The results show that 8 repeating ensures a good repeatability even for inaccurate models, thus this value is chosen for n_{eval}.

3.4 Training Algorithm

As the next step of the general workflow creation, a training algorithm has to be selected. During training, the weights and biases of the network are modified in order to minimize the error between the desired and actual outputs. The most common training method is the stochastic gradient descent (SGD) [16]. This algorithm is used to find a set of input parameters that results in a minimum of a target function. In case of neural network trainings, the input variables of the SGD are the weights and biases, and the target function is the loss function that describes the average prediction error for a subset (batch) of the training dataset. The SGD iteration follows the negative gradients of the loss function in order to find the minimum. The gradients are calculated with the back propagation algorithm. The degree of the change in the direction of the gradient is described by the step size (or learning rate). However, the selection of this hyperparameter is difficult because too large values result oscillations and the minimum cannot be found, while too low values cause slow convergence. In addition, the learning rate should be modified during the optimization because the step size should decrease as the minimum is approached. Therefore, the best practice is to use adaptive techniques that automatically change this hyperparameter during training.

In this section, six different adaptive training methods are investigated: the AdaGrad, RMSprop, Adadelta, Adam, Adamax and Nadam algorithms. The Adaptive Gradients (AdaGrad) algorithm is a simple SGD based method that uses an adaptive learning rate with respect to previous gradients [19]. The AdaGrad needs an initial learning rate, and later it calculates a step size for each dimension in the search space. The method is not that sensitive to the initial learning rate, thus 0.001 is used in this paper, which is a common default value. The Root Mean Squared Propagation (RMSprop) algorithm is based on the AdaGrad algorithm [20]. The problem of AdaGrad is that it can result too small step sizes at the end of the training. The RMSprop also calculates the step size from the previous gradients, but it uses a decaying average in order to eliminate the effects of early gradients, so the learning rate is mostly influenced by recent gradients. A gradient moving average decay factor (δ) with a value between 0 and 1 is required to determine the extent of the decay. Experience of researchers shows that the RMSprop is very efficient, and this is one of the best training algorithms for deep neural networks [16]. The Adadelta algorithm is based on the AdaGrad and RMSprop algorithms [21]. It also uses a gradient moving average decay factor to improve the influence of the recent gradients compared to the early ones like the RMSprop, but the step size is calculated differently. Another difference is that this method does not require an initial learning rate. The Adaptive Movement Estimation (Adam) algorithm is also a successor of AdaGrad and RMSprop [22]. This method uses a second decay factor; thus it has three hyperparameters. The commonly used default values (initial learning rate: 0.001; decay factor for first momentum: 0.9; decay factor for infinity norm: 0.999)

usually provide good results, thus this work operates with these. The Adam is a widely used ANN training method due to its good performance. In Adam, the weights are updated with the squared norm of past gradients. The AdaMax algorithm which is based on Adam, provides a more generalized approach as it uses the infinite norm of past gradients. The Nesterov-accelerated Adaptive Moment Estimation (Nadam) is another method based on Adam [23]. The main difference between the two algorithms, is that Nadam uses Nesterov's Accelerated Gradient for the calculations. This means that the weight updates are performed with the gradient of the projected update instead of the actual gradient. The default hyperparameter values also provide good results for Adamax and Nadam.

Similarly, to the previous section, the performance of the training algorithms will be represented by the validation R2 of the IMEP, PM and ID models. A repeated evaluation is used with 8 n_{eval} and the full evaluation time is recorded. Firstly, a good value for the δ of RMSprop and Adadelta is selected. Figure 3 presents the validation R2 for RMSprop and Adadelta with 9 different δ values. The decay factor of 0.95 shows a good compromise between the calculation time and the accuracy for both algorithms for all models, thus this is selected.

Now the performance of the six algorithms can be compared with each other at Figure 4. It is discernible that the fastest and most accurate methods are the Adam and the RMSprop for all models. The Nadam and Adamax models also have a good performance, but they are a bit slower. The investigated models have the same architecture for all six methods. The architecture can also have influences on the performance of the different methods thus this also has to be considered. However, this investigation demonstrated the superiority of Adam and RMSprop, hence only these methods are investigated next.

To study the performance of the Adam and RMSprop on different architectures, 90 models were analyzed with 1 and 2 hidden layers. The number of neurons in the layers was varied between 40 and 80, with a 5 neuron step, and each combination was evaluated with repeated evaluation. The average calculation time and validation R2 of both methods are presented at Table 1. The average calculation time decreased by 29.7% with the RMSprop method, while the accuracy remained the same. The PM model could be calculated 15.4% faster with the RMSprop, but the average validation R2 dropped by 0.6%. For the ID model, the achieved calculation time improvement with the RMSprop was 22.3%, and the accuracy decrease was 4.5%. Overall, the RMSprop can notably accelerate the training for the cost of a small drop in accuracy. Since multiple iterations are necessary to produce the final ANN model, the RMSprop is a better choice because it is also accurate enough. However, the Adam can also be used when the calculation time is not a limiting factor.

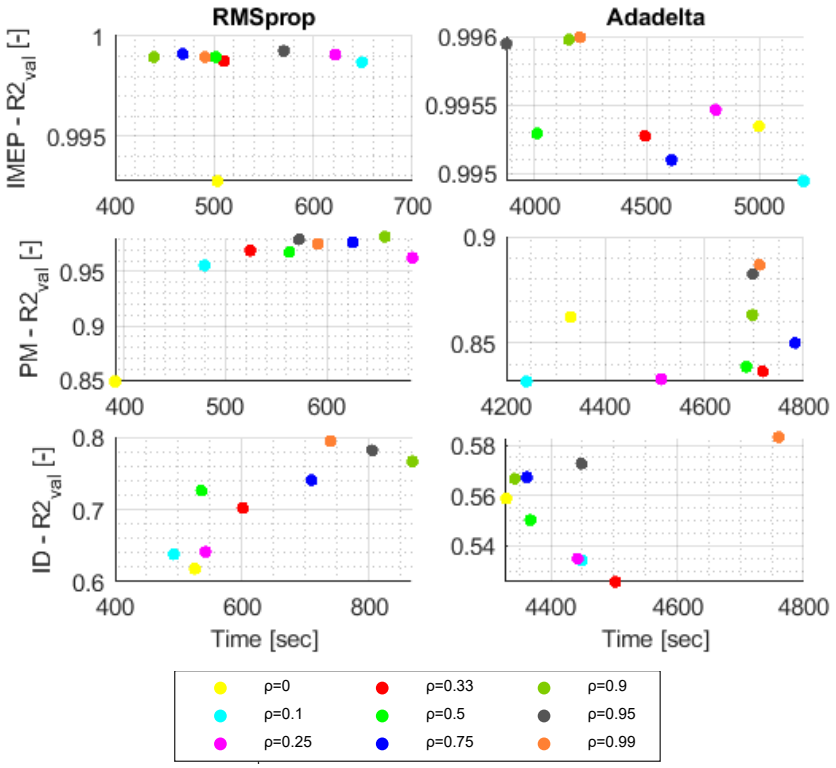

Figure 3
Performance of RMSprop and Adadelta with different moving average decay factors

Table 1
Average time and accuracy results for Adam and RMSprop algorithms

	***Adam* avg. calc. time [sec]**	***RMSprop* avg. calc. time [sec]**	***Adam* avg. val. R2 [-]**	***RMSprop* avg. val. R2 [-]**
IMEP	506.36	390.46	0.999	0.999
PM	690.20	598.13	0.980	0.974
ID	890.92	728.66	0.796	0.762

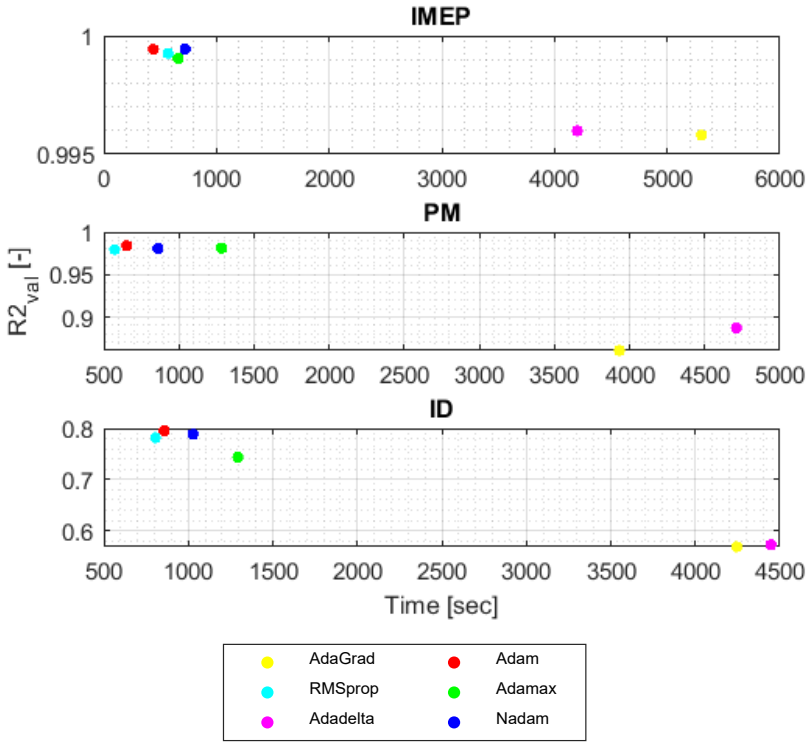

Figure 4
Comparison of the performance of six adaptive training methods

The training is an iterative method, where the samples of the training dataset are going through the network and the error of the result is used to update the weights of the model. The period when all samples participated in the weight update is called an epoch. Multiple epochs are necessary to create accurate ANN models, thus this is also a hyperparameter that has to be considered. If the number of epochs is small, the model will not be accurate enough (underfit). More epochs lead to more accurate models; however, overfitting can occur if this hyperparameter is too high. An overfitted model performs well in the training dataset, but it has a bad performance on the validation dataset. Moreover, the increased number of epochs leads to longer training time, so a good compromise has to be found. The early stopping method provides a good approach to use the proper number of epochs during training. A maximum epoch number is defined and when this is reached, the training stops. The training also stops if the validation MSE starts to increase. The stopping is not necessarily immediate: a patience parameter can be used, and the training only stops if the validation MSE increases continuously for a predefined number of epochs. This method avoids overfitting and reduces the training time. In this section, the optimal maximum number of epochs (ε_{max}) and the patience (p) hyperparameter is also investigated.

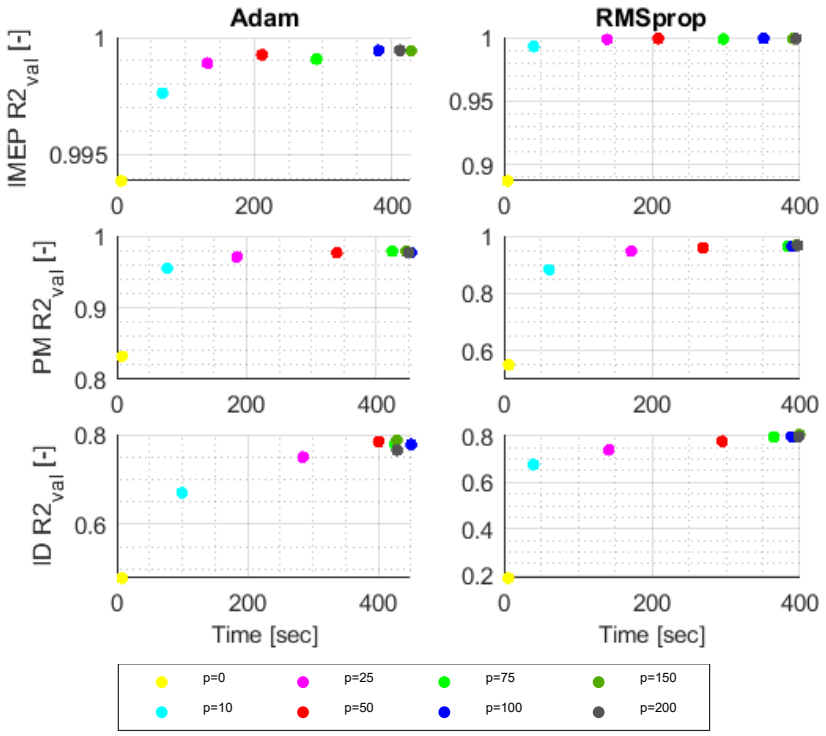

Figure 5
Performance of Adam and RMSprop with different patience values (max. number of epochs: 500)

The combination of 8 different p and 13 ε_{max} hyperparameters are investigated. First, a good p is selected. Higher patience results longer calculation time. However, the increase of accuracy stops when the ANN reaches a good fit. Therefore, the validation R2 converges to a certain level at each diagram of Figure 5. This figure shows that p=50 is a good choice for both Adam and RMSprop: this value usually provides the fastest training to reach the converged maximal accuracy level. There are some cases where lower p could be enough, but these lower values cannot provide good results for all models. Note that Figure 5 only presents the results with ε_{max}=500, which will be later chosen as the best ε_{max} value. All 13 ε_{max} was investigated and the results were similar for each: p=50 showed the best compromise, thus only one ε_{max} is presented here.

Next, the necessary ε_{max} is determined. The training can be stopped by two criteria: accuracy drop and reaching of ε_{max}. Usually, the best case if the training is stopped by the accuracy drop, because this means that a good fit was found and further training leads to overfitting. However, there can be some models that has too slow convergence, thus the training has to be stopped before reaching the best fit to reduce calculation time.

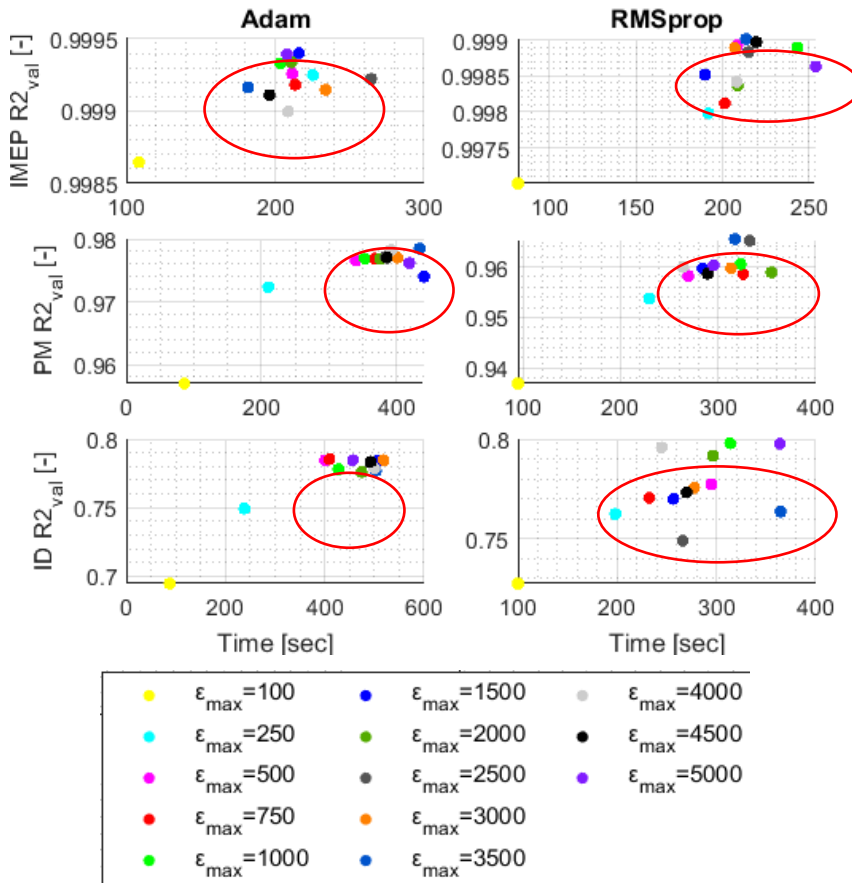

Figure 6

Performance of Adam and RMSprop with different maximum number of epochs (patience: 50 epochs)

Here, the aim is to find the lowest value for ε_{max} that provides enough epochs to develop good fit models for most cases. Figure 6 investigates the validation accuracy of the IMEP, PM and ID models when the patience is set to 50 epochs. The result of each diagram shows separate clusters with similar accuracies and training times. In these clusters, a good fit model was reached and the training was stopped by the accuracy drop. From here on, the ε_{max} has no influence on the calculation time and accuracy and the differences come from the randomness of the process. Note that the clusters of Adam algorithm are smaller, thus its results are more stable than the RMSprop. The best ε_{max} value is 500 epochs because this is the lowest that is present in all clusters.

To sum up the outcomes of this section, the RMSprop algorithm is the best from the investigated methods, if the training time is the bottleneck. Adam algorithm can also be used, when a little longer calculation time is acceptable and further

increase in accuracy is required. It is recommended to set the Adam algorithm's hyperparameters to their default values, and the δ of RMSprop to 0.95. Regarding early stopping, the 500 epoch ε_{max} and the 50 epoch p is suitable for both methods.

3.5 Input Feature Selection

Another important aspect of the ANN model creation is the selection of proper input features. Sufficient input information is needed to map a systems behavior. However, too much information can also worsen the accuracy since irrelevant data can lead the training into false paths. The two main approach for input feature selection are the supervised and unsupervised selection methods. The unsupervised methods ignore the outcome of the model, while the supervised methods use target variables to remove unnecessary features [24]. Supervised methods such as wrapper, filter or intrinsic methods usually provide better results. Wrapper methods create multiple models with different input features and select the useful ones. This provides really good results, but the computational costs can rise. Filter methods chose the important inputs with statistical scores between the input and output features. Intrinsic methods are built-in feature selection methods of some training algorithms. From these possibilities, the wrapper methods fit the best for our purpose, since high accuracy is required and the computational costs remain low due to the low number of available input features.

The recommended workflow is as follows. First, the possible influencing parameters have to be identified manually, based on the available information of the system. Then the quasi-constant features have to be removed, because these have no relevant data for the training. Then the redundant features also needed to be eliminated. Here, the priority of each feature can be predefined manually, and the higher priority can be held in the inputs. Next a wrapper method have to be performed. The recursive feature elimination (RFE) is such a method, where at first a machine learning algorithm creates models with all input features, and then the method starts to remove them. In this paper, such an RFE method is implemented to select the necessary input features from the preselected ones. The RFE also creates ANNs to select input features, thus the results include randomness. Therefore, the RFE process is also repeated 8 times, and the inputs that were among the results at least 50% of the repetitions are selected.

This workflow was implemented for the IMEP, PM and ID models. Since a new input feature set generates a modified behavior, a new architecture was selected using a grid search algorithm with the settings described in [15]. The investigated output features are combustion and emission relevant parameters. Therefore, the two main set of manually selected input features for the RFE are:

- the measured inlet properties: pressure, temperature, mass flow rate

- the measured mixture composition and formation relevant parameters: EGR valve position, engine speed, air-fuel equivalence ratio, inlet and outlet oxygen concentration, fuel dose, start of injection, number of injections, ratio of main injection compared to pre-injection

The RFE algorithm selected 4, 6, and 8 input features for the new IMEP, PM and ID models respectively. This behavior is logical because the harder the modelled problem, the more information is needed to properly map the system. Table 2 compares the test R2 and the test root mean square error (RMSE) of the new models with the original ones. The IMEP shows a decrease in accuracy, but it still has an excellent performance. The PM model has slightly better performance with the new input set. The ID model shows a drastic improvement. The original model was unacceptably inaccurate, but the new input features developed it into a well performing one. Overall, the used input feature selection method performed well, and can be included in our workflow.

Table 2
Comparing the performance of the IMEP, PM and ID models with the original and new input features

	IMEP		***PM***		***ID***	
	Original inputs	**New inputs**	**Original inputs**	**New inputs**	**Original inputs**	**New inputs**
$R2_{test}$	0.9995	0.9929	0.9874	0.9882	0.8303	0.9782
$RMSE_{test}$	0.078 bar	0.303 bar	0.130 g/kWh	0.120 g/kWh	1.103 °CA	0.413 °CA

3.6 Architecture Selection

The architecture of an ANN highly affects its performance. The capacity of a network describes the ability of learning complex problems. Generally, the more neurons and layers the network has, the higher its capacity. Too low capacities result underfit while too high capacities lead to overfit, hence a good architecture have to be identified. The grid-search algorithm is a common architecture selection method, where a lower and a higher boundary for the number of layers, and for the number of neurons in a layer is selected. Then all possible combinations are investigated between these boundaries with a certain step size and the architecture with the best accuracy is selected. This method is popular because of its simplicity and accuracy; however, the calculation time can be too high. We used this method in our previous research, but now faster have to be found due to its slow speed. The constructive architecture selection is a common technique to identify a good network topology [25]. This method uses an initial architecture that definitely provides too small capacity for the problem. Then, it adds new neurons and layers to the network to achieve a good fit. Such a simple constructive method was created in this work to replace the previous technique. First, it generates a model with a single layer that has an initial neuron number.

Then, it starts to increase this number with a defined step until it reaches an upper boundary. Next, it creates a new layer with an number of neurons corresponding to the step size, and it continues to increase this layer. The iteration stops when the defined maximal number of layers reached. However, the iteration usually does not last this long since it has an accuracy criterium that stops the process when fulfilled. This criterium investigates the validation R2, and when it reaches 0.98, the model is considered accurate enough [26]. If this accuracy cannot be reached with the defined topological boundaries, then the most accurate model is selected.

Table 3
Comparing the performance of the constructive architecture selection and the grid-search algorithm

	IMEP		***PM***		***ID***	
	Original method	**New method**	**Original method**	**New method**	**Original method**	**New method**
$R2_{test}$ **[-]**	0.998	0.998	0.977	0.980	0.759	0.740
$t_{archOpt}$ **[h]**	6.66	0.12	8.50	2.49	7.71	2.26

Table 3 demonstrates the architecture selection's calculation times ($t_{archOpt}$) and the achieved test R2 of the new constructive algorithm compared to the previous grid-search algorithm. The new test R2s did not change notably compared to the old method, but the improvement in calculation times is immerse. The excellent models can fit really fast, so the necessary time for architecture selection becomes small. The average and bad models also show about 70% improvement in the calculation time, thus the new method contributes to lower computing costs.

3.7 Measurement Grid Resolution Selection

The experimental investigation of new advanced fuels is a major cost of development due to the high price of special compounds. This can also raise the expenses of ANN development since the dataset is created by measurements. To reduce these costs, a measurement grid resolution selection method is also created in this paper. The measurement grid needs high enough resolution to provide sufficient data for the ANN training, but unnecessarily dense measurements have to be avoided to reduce fuel costs. More complex problems require a denser measurement to properly map the system's behavior. The correlation coefficient can describe the complexity of an input-output relationship. When R is close to 0 the two parameters are not related, when it's close to 1 the relationship is linear, while the intermediate values represent a nonlinear behavior. To identify the proper resolution, the R between the varied grid parameters and the target values of the investigation have to be determined and compared with the achievable accuracies of multiple models created with different resolution datasets.

To establish a best practice for resolution selection, the original high resolution dataset [15] is used that has 3 varied parameters in the measurement grid. This investigation examines the 10 target variable of [15] to identify optimal resolutions for different complexities. First, the complexity of the 30 input-output relationship is needed to be determined, and the correlation coefficient between these pairs will describe it (R_{pair}). This can be done with a parameter sensitivity measurement that requires a small amount of fuel to measure at least 10 sample per varied grid parameter. This 10 sample is recorded for each varied parameter in the predefined measurement range. The other varied parameters need to have a fixed value during the measurement to guarantee that the only influencer of the investigated outputs will be the investigated input feature. These fixed values are selected as the middle values in the range of the given input parameters. Then, the R_{pairs} can be calculated from the samples. This study calculated 30 R_{pairs} for the investigated 30 input-output relationships. Note, that the absolute values have to be used to describe the complexity of the relationships. Now, an optimal resolution has to be found for these $|R_{pair}|$ values.

In the next step, 72 datasets with different resolutions were created from the original high-resolution dataset. An ANN model was created with the workflow described in this paper for the 10 output feature with each dataset. Then a prediction was made with the created 720 ANN models for the 6618 samples of the original high-resolution dataset, thereby the performance of the models was explored with the most detailed information available on the system. The reported accuracy measure is the determination coefficient for the full dataset ($R2_{full}$). After the calculations, there are 72 $R2_{full}$ values for the 10 output features and the best model have to be selected for each output. A model is considered acceptable if its $R2_{full}$ is at least 0.98. If there are multiple models for an output feature that satisfies this criterium, the one that was created with the smallest dataset is selected as the best, since this requires the least fuel. Now, the optimal resolution of the 3 varied parameter of the measurement grid is known for each output feature. However, only 6 models provided results with $R2_{full}$ above 0.98, so only 18 data points can be used to describe the relationship between complexity and necessary resolution.

Figure 7 shows the 18 optimal resolution – |Rpair| data points. Here, the resolution means the number of grid points in the measurement range of a varied parameter. Three main parts can be separated. If |Rpair| is lower than 0.4, then the investigated input feature does not have a significant effect on the investigated target variable, therefore, a lower resolution is enough. The average resolution in this area is 6.5, thus 7 equidistant value is enough for the varied parameter.

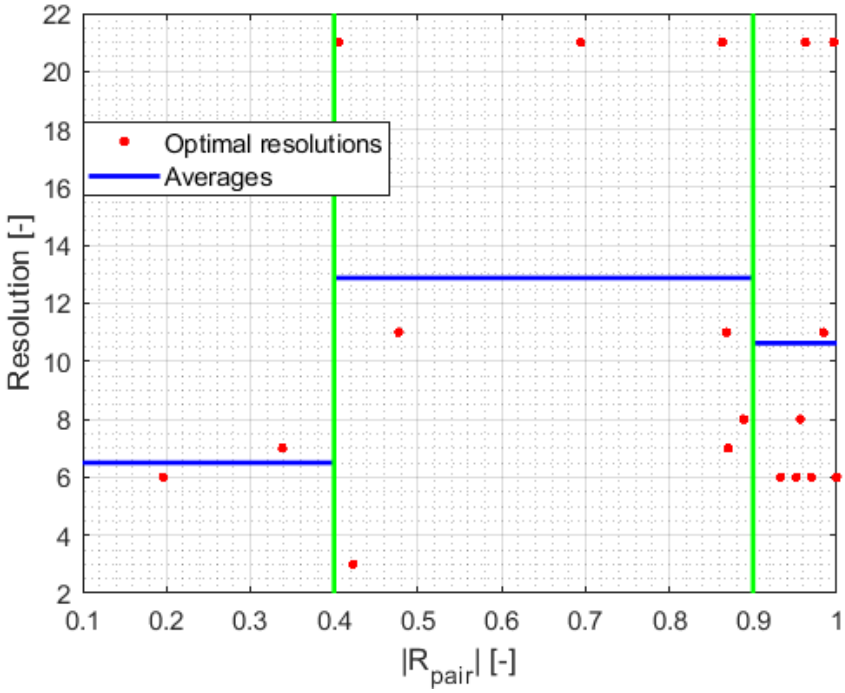

Figure 7
The optimal resolution of input-output pairs of different complexity

If |Rpair| is above 0.9, the correlation is strong between the varied parameter and the target variable, thus a higher resolution is needed to map the behaviour. The average is 10.625, so 11 variation is needed in this case. The in-between area needs an even higher resolution because here the relationship is highly nonlinear. The average resolution here is 12.87, so 13 equidistant value is necessary in the grid. This simple practice can create datasets with good resolutions to minimize fuel consumption while accurate models are provided. However, more investigation is needed to establish a more precise method for resolution selection.

3.8 Summary of the Created Methodology

The final workflow can be generated based on the previously presented investigations. Figure 8 summarizes the methods to efficiently create representative datasets from engine measurements, and to build accurate and fast ANN models from them. The upper part describes the methods related to the first goal. A proper measurement resolution can be designed that provides enough information to train the networks, while the fuel costs are minimized. This is based on a parameter sensitivity measurement, where the complexity of the

relationship between the varied grid parameters and the target variables are determined in order to select the proper measurement grid resolutions (Section 3.7). After the measurement, the created dataset is used to create a MLP type ANN. The lower part of Figure 8 demonstrates the workflow for this task with the steps and methods described previously (Section 3.2 – 3.6). The investigations proved that this workflow is able to reduce calculation time, fuel costs and provide accurate results.

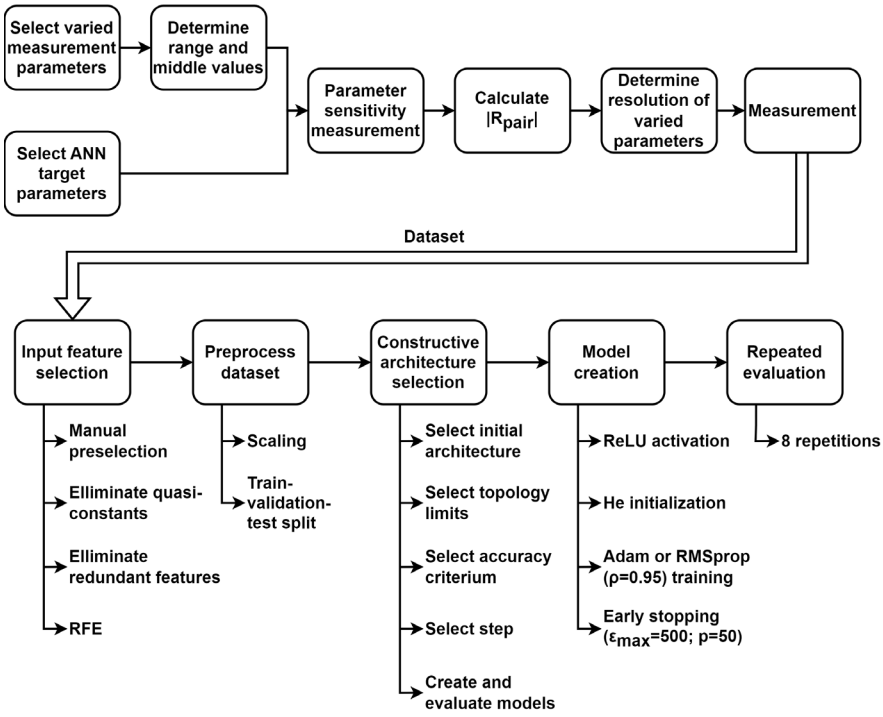

Figure 8
Final workflow to cost-efficiently create ANN models from engine measurements

Conclusions

This paper presented the investigations that led to the creation of a workflow for cost efficient ANN creation from engine measurements. A high resolution dataset was used to test multiple methods for the different steps of model generation. The results showed that the RFE method can be applied to select the necessary input features from a dataset and the constructive architecture selection is an efficient method to determine proper network structure. Regarding the training algorithms, the Adam and the RMSprop had the best performance from the investigated 6 method. The RMSprop is the recommended algorithm if the calculation time is the bottleneck. However, the Adam algorithm generates more accurate models. Therefore, this has to be applied if the speed of the training

process is not important. The random nature of the results has to be treated with repeated evaluation. The methodology also aids the determination of the proper measurement grid resolution. After a simple parameter sensitivity measurement, the complexity of the input-output relationships can be determined. Then, the resolution of the varied parameters can be selected.

The methodology ensures the cost-efficient creation of representative datasets and accurate ANN models; therefore, it contributes to help the development of our AI based e-fuel designer tool. The achieved accurate results prove that the AI is an important tool to enhance sustainable mobility. Besides our purposes, it can be applied to similar mathematical problems of different research fields as well. However, note that there is no general best practice for ANN creation, thus other researchers should consider the specialties of their mathematical problems.

Acknowledgement

The research leading to this result was funded by the KTI_KVIG_8-1_2021.

This work was supported by AVL Hungary Kft.

Abbreviations

AdaGrad, Adaptive Gradients; Adam, Adaptive Movement Estimation; AI, Artificial Intelligence; ANN, Artificial Neural Network; BP, Back Propagation; EGR, Exhaust Gas Recirculation; EU, European Union; ICE, Internal Combustion Engine; ID, Ignition Delay; IMEP, Indicated Mean Effective Pressure; MAE, Mean Absolute Error; MLP, Multilayer Perceptron; MSE, Mean Squared Error; MSLE, Mean Squared Logarithmic Error; Nadam, Nesterov-accelerated Adaptive Moment Estimation; PM, Particulate Matter; ReLU, Rectified Linear Unit; RFE, Recursive Feature Elimination; RMSE, Root Mean Square Error; RMSprop, Root Mean Squared Propagation; SGD, Stochastic Gradient Descent; STD, Standard Deviation; WTW, Well-to-wheel;

References

[1] P. Plötz, J. Wachsmuth, T. Gnann, F. Neuner, D. Speth, S. Link: Net-zero carbon transport in Europe until 2050 – targets, technologies and policies for a long-term EU strategy, Fraunhofer Institute for Systems and Innovation Research ISI, 2021

[2] M. Ovaere, S. Proost: Cost-effective reduction of fossil energy use in the European transport sector: An assessment of the Fit for 55 Package, Energy Policy, 2022, Vol. 168, 113085

[3] Zalacko, R., Zöldy M., and Simongáti Gy: "Comparative study of two simple marine engine BSFC estimation methods." Brodogradnja: Teorija i praksa brodogradnje i pomorske tehnike 71.3 (2020): 13-25

[4] A. Vučetić, V. Sraga, B. Bućan, K. Ormuž, G. Šagi, P. Ilinčić, and Z. Lulić, “Real Driving Emissionsfrom Vehicle Fuelledby Petrol and Liquefied Petroleum Gas (LPG)”, CogSust, Vol. 1, No. 4, Sep. 2022

[5] T. Koller, C. Tóth-Nagy, and J. Perger, “Implementation of vehicle simulation model in a modern dynamometer test environment”, CogSust, Vol. 1, No. 4, Sep. 2022

[6] eFuel Alliance: Press Release, 2022. Only available at: https://www.efuel-alliance.eu/fileadmin/Downloads/E_PM_eFA_EU-Parlament_vor_Richtungsentscheidung_02062022.pdf

[7] M Zöldy, "The effect of bioethanol–biodiesel–diesel oil blends on cetane number and viscosity." 6th international colloquium, January. 2007

[8] J. Matijošius, A. Juciūtė, A. Rimkus, and J. Zaranka, “Implementation of vehicle simulation model in a modern dynamometer test environment”, CogSust, Vol. 1, No. 4, Sep. 2022

[9] E. Cipriano, T. C. F. da Silva Major, B. Pessela, A. A. Chivanga Barros: Production of Anhydrous Ethyl Alcohol from the Hydrolysis and Alcoholic Fermentation of Corn Starch, Cognitive Sustainability, 2022, 1(4) https://doi.org/10.55343/cogsust.36

[10] M. Zöldy, P. Baranyi: The Cognitive Mobility Concept, Infocommunications Journal, 2023

[11] M. Zöldy, "Improving heavy duty vehicles fuel consumption with density and friction modifier." International Journal of Automotive Technology 20 (2019): 971-978

[12] M. Jadidi, S. Kostic, L. Zimmer, S. B. Dworkin: An Artificial Neural Network for the Low-Cost Prediction of Soot Emissions, Energies, 2020, Vol. 13, 4787, https://doi.org/10.3390/en13184787

[13] X. H. Fang, F. Zhong, N. Papaioannou, M. H. Davy, F. C. P. Leach: 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, Vol. 23(7), pp. 1201-1212

[14] Y. Cui, H. Liu, Q. Wang, Z. Zheng, H. Wang, Z. Yue, Z. Ming, M. Wen, L. Feng, M. Yao: Investigation on the ignition delay prediction model of multi-component surrogates based on back propagation (BP) neural network, Combustion and Flame, 2022, Vol. 237, 111852

[15] M. Virt, M. Zöldy: Artificial Neural Network Based Prediction of Engine Combustion and Emissions from a High Resolution Dataset, 1st IEEE International Conference on Cognitive Mobility, 2022

[16] I. Goodfellow, Y. Bengio, A. Courville: Deep Learning, MIT Press, 2016

[17] X. Glorot, Y. Bengio: Understanding the difficulty of training deep feedforward neural networks, Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, PMLR 9:249-256, 2010

[18] K. He, X. Zhang, S. Ren, J. Sun: Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, Proc. IEEE Int. Conf. Comput. Vis. (ICCV), pp. 1026-1034, 2015

[19] J. Duchi, E. Hazan, Y. Singer: Adaptive Subgradient Methods for Online Learning and Stochastic Optimization, Journal of Machine Learning Research, Vol. 12, pp. 2121-2159, 2011

[20] T. Tieleman, G. Hinton: Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, COURSERA: Neural Networks for Machine Learning, 2012

[21] M. D. Zeiler: ADADELTA: An Adaptive Learning Rate Method, arXiv preprint arXiv:1212.5701, 2012

[22] D. P. Kingma, J. L. Ba: ADAM: A Method for Stochastic Optimization. https://arxiv.org/abs/1412.6980, 2018

[23] T. Dozat: Incorporating Nesterov Momentum into Adam, Proc. ICLR, pp. 1-4, 2016

[24] M. Kuhn, J. Kjell: Applied predictive modelling, New York: Springer, vol. 26, 2013

[25] S. Curteanua, H. Cartwright: Neural networks applied in chemistry. I. Determination of the optimal topology of multilayer perceptron neural networks, J. Chemometrics, Vol. 25, pp. 527-549, 2011

[26] S. Dey, N. M. Reang, P. K. Das, M. Deb: Comparative study using RSM and ANN modelling for performance-emission prediction of CI engine fuelled with bio-diesohol blends: A fuzzy optimization approach, Fuel, Vol. 292, 2021