

Real time estimation of emissions in a diesel vehicle with neural networks

Teresa Donateo^{1,*}, Riccardo Filomena¹

¹Università del Salento, Dipartimento di Ingegneria dell'Innovazione, via per Monteroni, 73100, Lecce, Italy

Abstract. Several studies in literature have shown how real-world emissions strongly depend on driving condition, driving style, ambient temperature and humidity, etc. so that they are significantly different from the values measured on test benches over standard driving cycles. This concern, together with the so-called Diesel-gate, has caused the introduction in Europe of an innovative procedure for the registration of vehicle based on real driving emissions (RDE) measured with a portable emission measurement system (PEMS). PEMS devices are bulky and very expensive, therefore they cannot be extensively for an actual real time monitoring of emissions. To solve this problem, the present work proposes a Neural Network model based on the interpolation of the time-histories of driving conditions (speed, altitude, ambient temperature, humidity and pressure) and emissions measured on a diesel start-and-stop vehicle while performing a series of RDE tests. Two different approaches are proposed. The first one calculates the emissions on the basis of the vehicle motion (speed and altitude profile, ambient conditions). The second one models the engine block using as input the ambient conditions, the load and the rpm of the engine as derived from the OBD-II scanner. The output of both models are the flow rates and cumulated values of CO₂ and NO_x. Note that the inputs of the two models are signal that can easily obtained on-board without additional sensors.

1 Introduction

Increasingly stringent emissions regulations are threatening the future of diesel. On the one hand, it enables significant savings of fuels and, therefore, of greenhouse emissions, in long trips. On the other hand, the diesel engine is no longer viewed favourably in public opinion and by European legislators who consider this engine to be particularly polluting.

European standards define acceptable exhaust emission limits for vehicles sold in the European Union [1] - [3]. They are defined by a series of directives that stage the gradual introduction of increasingly stringent limits. At present, the standard to be met appears are: Euro 6 and US EPA 2010 for heavy vehicles (HDV1) and Euro 6 and Tier 2 for light vehicles (LDV2). With the new European provisions RDE (Real Driving Emissions) a further tightening has been introduced which aims at a substantial block for diesel cars from 2020.

* Corresponding author: teresa.donateo@unisalento.it

In order to obtain pollution data as real as possible, emissions are to be measured under real road driving conditions by means of special instruments installed on board called PEMS (Portable Emissions Measurement System) [4] - [10]. PEMS devices are bulky and very expensive, therefore they cannot be extensively for an actual real time monitoring of emissions. The goal of this investigation is to verify if it is possible to predict, with reasonable accuracy, the overall emissions of NO_x and CO₂ during an RDE cycle with a numerical model using as inputs signals already acquired in a vehicle.

In the literature, it is possible to find numerous studies carried out in the laboratory with the ultimate aim of generating models to predict emissions. One of the greatest challenges lies in the difficulty to characterize the system behaviour both in stationary and transient conditions [11],[12].

The strong dynamic component of the problem has seen the use of various types of neural networks, from the simplest Feed Forward to the more complex Recurrent [13]-[15]. The results, of what appears to be the state of the art, are around 80-90% accurate [16]-[18].

In this investigation, we will use real CO₂ and NO_x emissions measured in RDE tests to develop Neural Network models. Two different approaches are proposed. The first one calculates the emissions from the vehicle motion (speed and altitude profile, ambient conditions) and is aimed at giving the driver an estimation of the emissions produced in a RDE test. The second one, models the engine block using as input the ambient conditions, the load and the rpm of the engine derived from the OBD-II scanner and is meant to be used in the framework of hybrid electric vehicles.

Even if the tools and the topics addressed in this study are not particularly innovative, it is new the application to driving cycles complying the RDE legislations. Moreover, this investigation is only the first step towards the development of advanced energy management strategies for hybrid electric vehicles aimed at reducing CO₂ and NO_x.

2 The vehicle and the RDE cycles

The experimental data used in this investigation were acquired on a Class 3b vehicle (maximum speed >120 km/h, power-to-unladen-mass ratio > 34 W/kg) whose specifications cannot be reported because of a confidentiality agreement. The vehicle was type approved in 2013 and had a mileage of 70000 km when the tests began. It was equipped with an AVL PEMS 493 as shown in Fig. 1. This unit includes multiple gas analysers, a GPS receiver (to record vehicle speed, latitude, longitude and altitude), an exhaust flow meter and an interface for connection to the vehicle's On-Board Diagnostics (OBD). Ambient temperature, humidity and pressure are measured using appropriate sensors.

Several RDE cycles (each with a duration between 93 and 108 minutes) were performed in Lecce with this equipment. Data were sampled with a frequency of 10 Hz resulting in a total of more than 570000 samples (> 15h). For the details about the tests and their outcomes, please refer to [4].

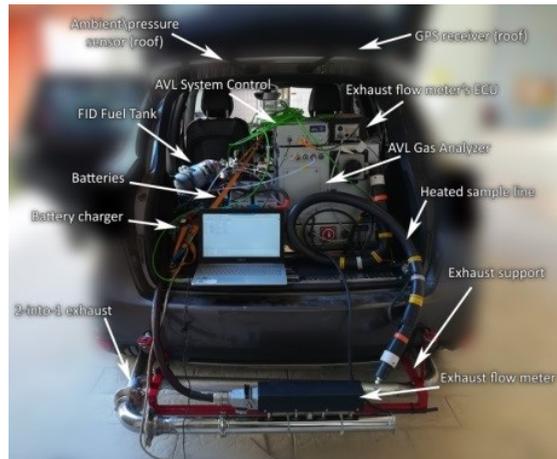


Fig. 1. The vehicle equipped with the PEMS instrumentations

2.1 Pre-processing of the raw data

A wise selection of data and the subsequent pre-processing stage are crucial for a proper training phase. In this work, we selected the following data vectors as predictors due to their low Pearson coefficient value:

- Air temperature (°C);
- Air pressure (mbar);
- Altitude (m);
- Vehicle speed (km/h);
- Calculated engine load (%);
- Engine speed (rpm);
- Exhaust gas flow (l/h);
- CO₂ (g/s);
- NO_x (g/s).

2.1.1. Outlier and smoothing filterd

Taking account of data distribution, which does not fit a Gaussian model, we used moving median method as outlier detection strategy. That method substitutes anomalous values by interpolating the previous non-outlier to the next one.

A smoothing filter was applied to highlight significant patterns to attenuate the noise generated by several phenomena (environmental, electrical, computer artifacts or by high speed). That filter employs moving average method calculated with respect to a specified time window span. Moreover, as a side effect, this routine reduces the training time as shown by experimental evidence.

2.2 Traction force and traction power

Considering the goal of this investigation, we decided to generate traction force and traction power data vectors. In this way, we created a wide ranged model linked to variables like vehicle acceleration, mass and size specifications. The following approach, we adopted, (by

Guzzella et al (2007) [19]) uses the elementary equation describing the longitudinal dynamics of a road vehicle:

$$m_v \frac{d}{dt} v(t) = F_t(t) - (F_a(t) + F_r(t) + F_g(t) + F_d(t))$$

where m_v , $v(t)$ are the mass and speed of the vehicle; F_t is the traction force; F_a is the aerodynamic friction; F_r is the rolling friction; F_g is the grade force; F_d is the disturbance force that summarizes all other not yet specified effects. F_t represents the force generated by the main engine minus the force used to accelerate the rotating parts inside the vehicle and minus all the friction losses of the powertrain. Assuming the system works in ideal conditions, this force is completely transferred to the wheels as power:

$$P_w = F_t \cdot v$$

Both generated signals are function of: mass, acceleration, frontal area A_f , aerodynamic drag coefficient c_d , rolling friction coefficient c_r .

2.3 Data Scaling

Right after filtering data, they have been scaled to fit the range required by the input neurons in the neural network. This task has been achieved by normalizing the signals between 0.1 and 0.9, computing them as equation below shows:

$$X_{i,new} = \frac{X_i - X_{min}}{X_{max} - X_{min}}$$

Table 1 shows all the data that will be used for the training of the neural network.

Table 1. Selected data

Variable	Unit
Ambient Temperature	C
Ambient Pressure	mbar
Altitude	m
Vehicle speed	km/h
Engine load	%
Engine speed	rpm
Traction force	kN
Traction power	kW
CO ₂	g/s
NO _x	g/s

3 Development of the neural networks

We developed two different models using the same variable as target of forecast as shown in Fig. 2. The “Vehicle” model was designed to be an auxiliary cockpit during RDE test while the “Engine” model will be implemented on a hybrid powertrain model.

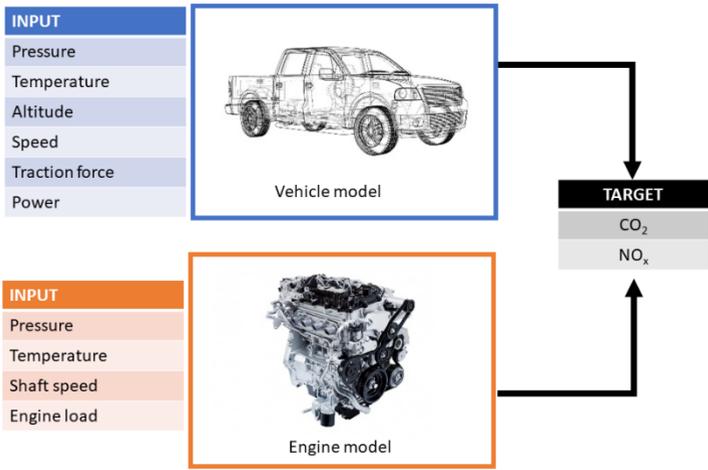


Fig. 2. Inputs and targets of the proposed models

3.1 Multi-layer Perceptron

An Artificial Neural Network (ANN) is a dynamic system having oriented graph as topology (Fig. 3). Each node is called artificial neuron, while arcs are defined as synaptic weight. The term network come from the layered topology connecting neurons.

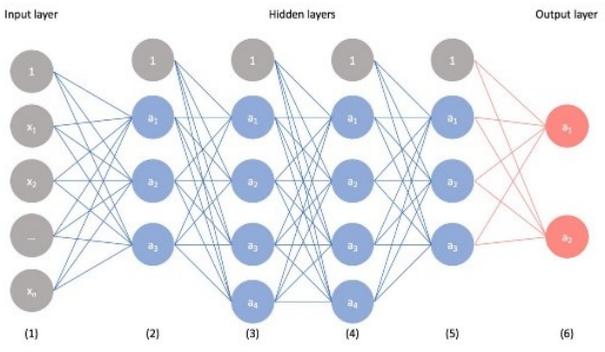


Fig. 3 Architecture of a multi-layer perceptron neural network.

It can be seen as a black box that associates an input to an output. Taking advantage of that, a neural network become a mathematical function $f(input; weights)$ whose behaviour varies according to the weights and of course the input, without specifying the shape of the f function. The present study aims at generating a light and responsive model for a real time forecasting of emission. For this reason, we chose a multilayer perceptron feedforward ANN that is characterized by a simple structure. The chosen optimization algorithm, Adam, is an alternative to the classic SGD that maintains a different learning rate for each parameter, combining the advantages of RMSProp and AdaGrad. Moreover, it operates in a computationally efficient way [20].

3.2 Training options

The whole dataset has been divided into three main parts:

- 70% training set;
- 15% evaluation set;
- 15% test set.

However, the distribution of the tests in the training, evaluation and test sets is a critical issue that will be analysed as a further investigation. As for the structure, two different configurations have been chosen, one with single and double hidden layers. This choice was made because of the desire to achieve the highest possible degree of generalization. In fact, a too complex structure could "learn" too much from training data and show bad results in the test phases. A Dropout logic has also been inserted to minimize possible Overfitting problems. Another discriminating factor taken into consideration is the activation function chosen for the single layers. We opted for a sigmoid and a non-linear ReLU activation function. The error functions taken into consideration were:

- $RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - h_{\theta}(x_i))^2}$;
- $MAE = \frac{1}{n} \sum_{i=1}^n |y_i - h_{\theta}(x_i)|$;
- $L_{\infty} = \max_i |y_i - h_{\theta}(x_i)|$.

Where y_i are the target values from the RDE tests and $h_{\theta}(x_i)$ are the predicted values by the ANN. In particular, the last error function is called *infinite norm* or *Chebyshev distance*. It appears to be more robust in controlling residual errors, but, on the other hand, it can have a strong sensitivity in the presence of noise on the signal [21].

To sum up, the hyperparameters taken into consideration are shown in the Table 2.

Table 2. Hyperparameters selected.

Hyperparameter	Range
Layer	{1,2}
Neurons	[3,20]
Dropout	[0,1]
Activation Function	{Sigmoid, ReLU}
Learning Rate	[0.001, 0.05]
Loss Function	{RMSE, MAE, L_{∞} }

The goodness-of-fit of the trained ANN has been evaluated by using three statistic estimators: RMSE, coefficient of determination (R^2) and cumulative relative error.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - h_{\theta}(x_i))^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

$$Err_{cum} = \frac{\sum_{i=1}^n (y_i - h_{\theta}(x_i))}{\sum_{i=1}^n y_i}$$

3.3 Manual optimization

The first step was to identify the best error function among the available ones. A reduced number of combinations was used with a particular focus on loss functions. The details are in Table 3 .

Table 3. First step hyperparameters selected.

Hyperparameter	Range
Layer	1
Neurons	[3,10]
Dropout	0
Activation Function	ReLU
Learning Rate	[0.001, 0.05]
Loss Function	{RMSE, MAE, L_{∞} }

The basic configuration is therefore a single hidden layer, with variable number of neurons, a fixed range for LR and ReLU as activation function and no dropout. Networks have been trained for all available combinations sorted according to the desired model (Engine and Vehicle), target emission (CO_2 and NO_x) and loss function. Tests have shown that the MAE function offers the best performance in almost all cases covered and, in particular, in minimizing the cumulative error. The plots of Fig. 4 show the trend of the target function as the number of neurons and the learning rate changes for one of the models examined in this study.

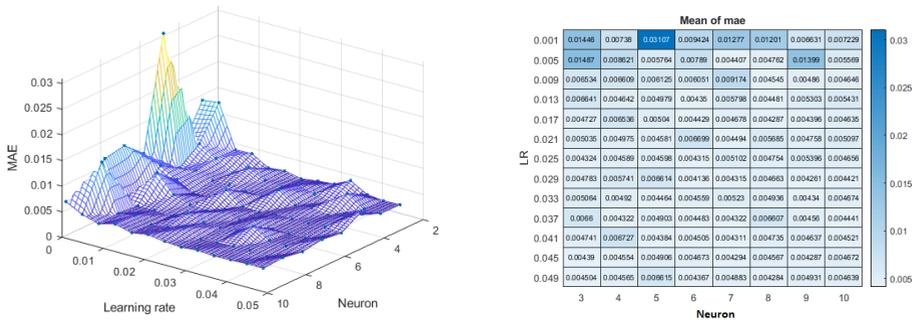


Fig. 4. Vehicle Model (NO_x): 3D surface (left) and heat map (right) of the MAE function as the learning rate and number of neurons vary.

3.4 Bayesian optimization

In the previous paragraphs, experiments on the models were carried out using manually chosen values as hyperparameters. These values are the result of numerous tests whose objective was to improve the result as much as possible and therefore the minimization of the loss. What makes such an approach difficult is that, for each model, the optimum requires its own configuration, not necessarily the same as the others, even if using the same data. The time required to achieve an optimal configuration could be considerable or even not lead to concrete results, but thanks to specific optimizers, the operation is easier and immediate.

The Bayesian optimization approach uses mathematical models and does not require human intervention at any stage of research. The tool that has been used combines the assisted approach to the algorithmic one, delegating to the programmer only the choice of the value range for the hyperparameters and automating the search through an algorithm called Bayesian Optimizer. Bayesian optimization maintains a Gaussian process model of the target function internally and uses its evaluations to train the model. An innovation in Bayesian optimization is the use of an acquisition function, which the algorithm uses to determine the next point to consider. The acquisition function can balance sampling in areas that have not yet been well modelled and, therefore, have poor performance of the target functions [22] [23]. Unlike the random or grid search, therefore, the results of past evaluations are tracked to form a probabilistic model that maps the hyperparameters according to the probability of a score on the target function [24]. In literature, this model is called "surrogate" for the objective function and is represented as $p(y|x)$ [25]. The latter is much easier to optimize than the objective function. The Bayesian optimization looks for the next set of hyperparameters to evaluate on the real target function by selecting those that work best on the surrogate function. The training was carried out on a computer with the following technical specifications: CPU AMD Ryzen 3 2200g, GPU integrated AMD Vega 8, RAM 16Gb (2x8) 3200 MHz. Table 4 shows the average convergence times categorized by model, target and number of layers.

Table 4. Average convergence times (seconds).

	ENGINE		VEHICLE	
	NO _x	CO ₂	NO _x	CO ₂
Single Layer	30.3 sec	15.3 sec	31.1 sec	20.3 sec
Double Layer	44.5 sec	25.2 sec	45.9 sec	28.9 sec

4 Discussion of the results

The networks were trained through Bayesian algorithmic logic differentiating by model (Engine and Vehicle), target (CO₂ and NO_x) and number of layers (single and double). In Fig. 5-Fig. 7 the results of the tests are summarized.

The optimal structure of the neural network varies significantly for the different cases, as can be seen from the plots. The ReLU activation function tends to have the worst results in almost all metrics and the only exceptions are the NO_x and CO₂ cumulative error in the single-layer engine model (fig. 3.19). The sigmoid function offers better performance in all other case studies.

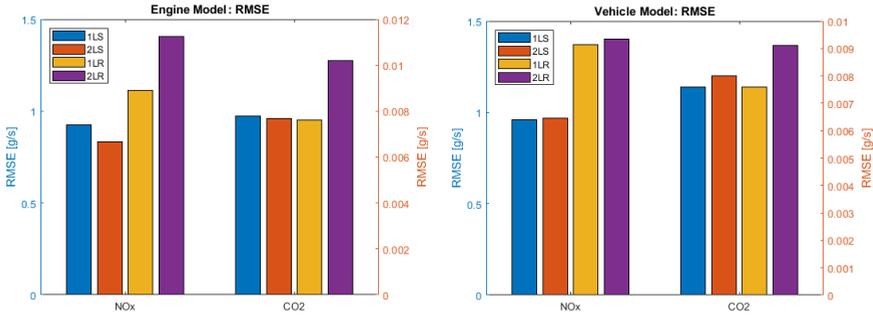


Fig. 5. RMSE values for Engine and Vehicle model, for NO_x and CO₂ as the number of layers (1L, 2L) and the activation function (S, R) vary. A low value indicates better performance.

The single layer configuration gave better results in predicting NO_x, whereas the CO₂ target prefers a double layer structure. In general, the CO₂ target, as expected is easier to predict by showing better R^2 and fewer cumulative errors (fig. 3.19). Finally, we can say that the Engine model is better predicted by neural networks in terms of R^2 (fig. 3.18), but we obtain smaller cumulation errors with the Vehicle model (fig. 3.19).

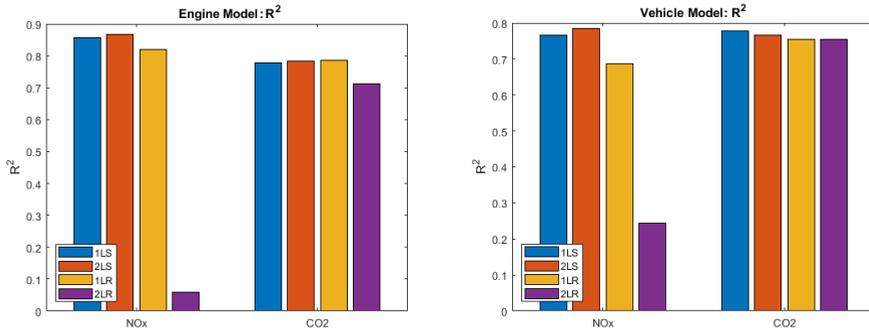


Fig. 6 R-squared values for Engine and Vehicle model, for NO_x and CO₂ as the number of layers (1L, 2L) and the activation function (S, R) vary. A high value indicates better performance.

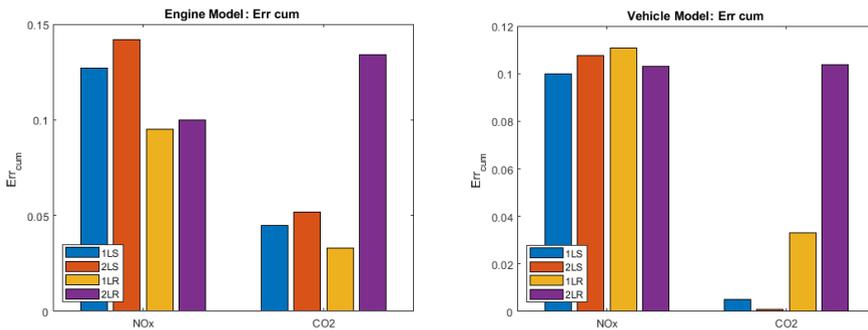


Fig. 7 Err_{cum} values for Engine and Vehicle model, for NO_x and CO₂ as the number of layers (1L, 2L) and the activation function (S, R) vary. A low value indicates better performance.

The last step of this process was the selection of the best combination of hyperparameters with respect to the chosen metrics, giving priority to the accumulation error. Table 5 shows the combinations of hyperparameters of the selected neural networks with respective results.

Table 5. Summary of the best models with their parameters.

Engine - NOx		Engine – CO2		Vehicle - NOx		Vehicle – CO2	
Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
Layer	1	Layer	2	Layer	1	Layer	2
Neuron (L1)	20	Neuron (L1)	6	Neuron (L1)	5	Neuron (L1)	6
Dropout (D1)	0.3	Dropout (D1)	0.2	Dropout (D1)	0.0	Dropout (D1)	0.2
Neuron (L2)	-	Neuron (L2)	5	Neuron (L2)	-	Neuron (L2)	7
Dropout (D2)	-	Dropout (D2)	0	Dropout (D2)	-	Dropout (D2)	0.5
LR	0.026	LR	0.024	LR	0.042	LR	0.05
Activation	ReLU	Activation	Sigmoid	Activation	Sigmoid	Activation	Sigmoid
Loss	MAE	Loss	MAE	Loss	MAE	Loss	MAE
RMSE	0.0073 [g/s]	RMSE	0.69 [g/s]	RMSE	0.0066 [g/s]	RMSE	0.927 [g/s]
R-squared	0.861	R-squared	0.878	R-squared	0.846	R-squared	0.804
Err. Cum	0.67%	Err. Cum	0.34%	Err. Cum	0.53%	Err. Cum	-0.43%

Note that we managed to minimize the error on accumulation to a value below 1% and to obtain R^2 around 85%. This gives us both precision on the measurement of pollutants, the objective of this work, and a high understanding of the local signal trend. The latter is very important whenever a strategy to reduce emissions is to be structured. In fact, capturing the positive gradients of CO_2 and NO_x in real time allows a greater reactivity of the control algorithms.

In general, the number of neurons of the proposed Neural Networks is low, which emphasizes a greater ability to predict targets by simpler models. The result is not surprising, as simpler models tend more to generalize and get better results in the test phase. Finally, it should be noted that, in all combinations, the dropout factor has remained around zero, which means that the early stopping parameters were set correctly, and overfitting problems were rarely found during training. The instantaneous and cumulated emissions of NO_x along a test are shown in Fig. 8 together with the speed profile for the vehicle model. Similar results are obtained in for NO_x and for CO_2 with the engine model. Note that, despite the final value of the cumulate is practically identical, the instantaneous values of NO_x are not accurately predicted, above all in the rural and motorway sections of the cycles. This is probably due to an uneven distribution of the amount of data in relation to the operating sections. The RDE test provides the same number of km for each zone, but at different speeds. It is therefore obvious that higher speeds (rural, motorway) result in less run time and, consequently, fewer samples. The local accuracy issues are therefore the result of a particularly pronounced under fitting.

To improve the results, the procedure is being applied separately to each section as a further investigation. A preliminary result of this study is also shown in Fig. 8 where the results of the Neural Network trained and tested only on the urban sections of the RDE tests is reported as a black line. Note the significant improvement that can be seen qualitative in the capability to follow the peaks and the valleys of the NO_x experimental curve and in the cumulate trace. Quantitatively, the RMSE decreases from 0.0028g/s to 0.0022g/s, the R^2 increases from 0.56 to 0.67, the error on accumulation on the urban section decreases from 16% to 0.85%, by training the model on the urban section only. This proves that by training the neural network on consistent data the algorithm improves its prediction capabilities.

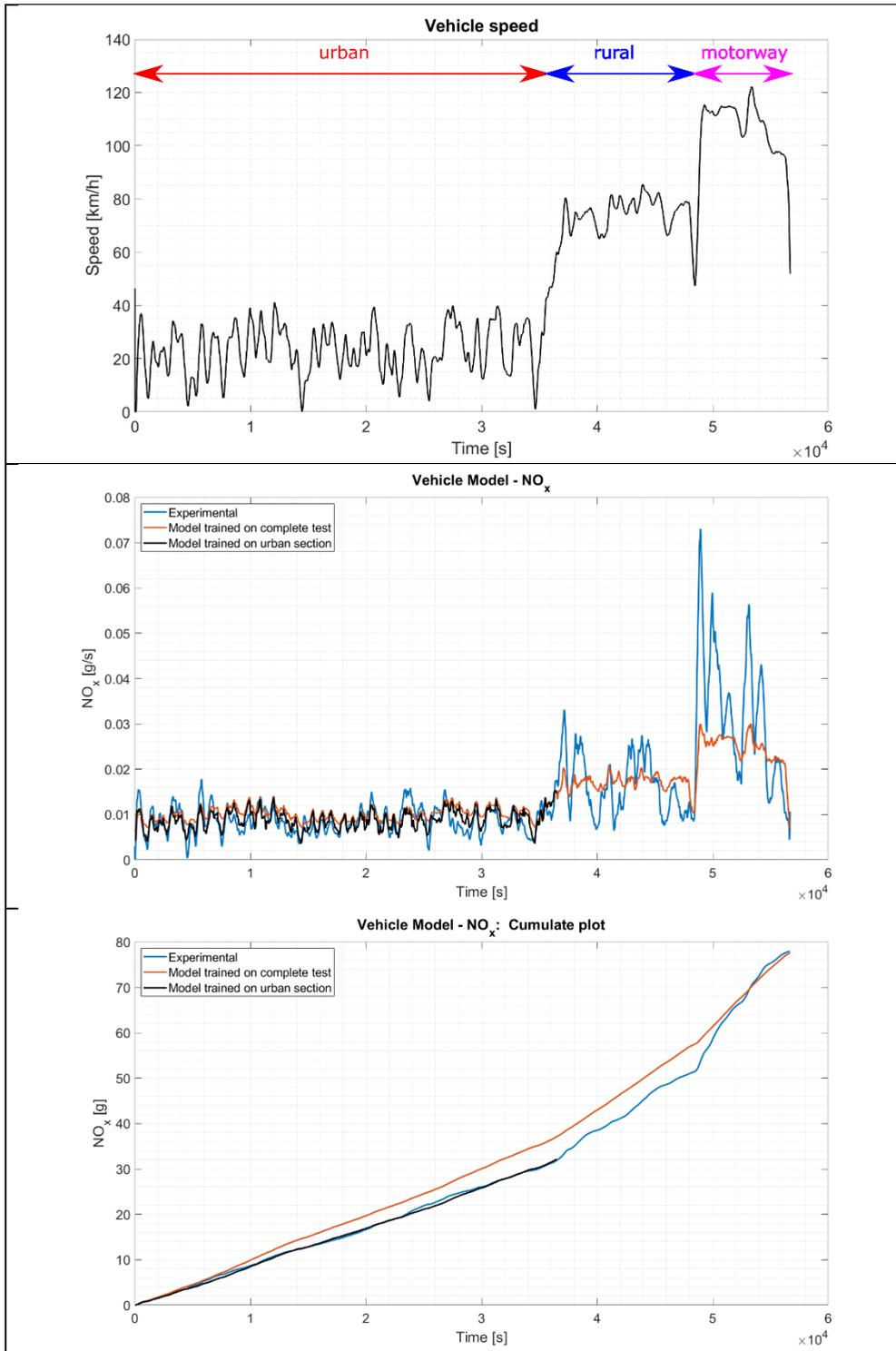


Fig. 8. Engine model - Comparison of instantaneous NO_x between the experimental data (blue) and the predicted curve obtained with the neural network (orange). Black lines show the preliminary results of the training performed on the urban section only

CONCLUSIONS

This study describes the modelling with Neural Network of real-world emissions of NO_x and CO_2 in a diesel vehicle using experimental data obtained with a Portable Emissions Measurement System on a Diesel start and stop vehicle of class 3b during RDE tests. The first phase of the investigation was the pre-treatment of the dataset in Matlab environment. The following data of interest were selected: temperature and atmospheric pressure, vehicle speed, altitude, engine rpm, engine load, NO_x , CO_2 . In addition, the vectors of traction force and power to the wheels were generated to address the dynamic effects. A first filter of outliers and one of smoothing then eliminated part of the noise present on the signals. The last phase of pre-processing saw the normalization of all signals in a range between 0.1 and 0.9. Two different models (Engine and Vehicle) were created which had different input data, but the same targets for the prediction. The main objective of the investigation was to minimize the cumulative error on emissions and, in addition, other metrics such as RMSE and R^2 were considered. The basic model chosen for the neural network was Perceptron with a multi-level structure. The hyperparameters related to it were treated in detail and the possible combinations of these were analysed. In particular, the performance of three different error functions were evaluated: MAE, RMSE, infinite norm. The MAE showed significantly better results in the test phase, and was therefore selected for the training. To deal with the high number of combinations, an algorithm-based optimization logic was used. The Bayesian optimizer, implemented in Matlab environment, creates an internal model called surrogate that, iteration by iteration, simulates the behaviour of the loss function. This allowed a more careful choice of hyperparameters, minimizing the number of combinations examined by the model and significantly reducing the computational time. The results shown that simpler models, single layer and low number of neurons, guarantee better performance. Moreover, the Sigmoid activation function is able to describe physical phenomena with greater accuracy. In conclusion, the trained networks achieved performance with cumulative errors of less than 1% and R^2 around 85%, both for the Vehicle and Engine models and for CO_2 and NO_x emissions. Nota that the Vehicle model is designed to be a sort of auxiliary "cockpit" with the aim of providing the driver with an estimate of emissions on a certain route or as an aid to the pilot during the RDE tests. The Engine model, instead, has, as future development, the development of advanced energy management strategies for hybrid electric powertrains. In this way, it will be possible to evaluate the effects of hybridisation in terms of both CO_2 and NO_x .

SYMBOLS AND ABBREVIATIONS

c_d = aerodynamic drag coefficient;

c_r = rolling friction coefficient;

Err_{cum} = cumulative relative error;

F_a = aerodynamic friction;

F_d = disturbance force;

F_g = force due to non-horizontal roads;

F_r = rolling friction;

F_t = Traction Force;

h_θ = predicted values by the ANN;

L_∞ = Infinite Norm;

m_v = mass of the vehicle;
 P_w = Power at the wheels;
 R^2 = Coefficient of Determination.
 v = speed of the vehicle;
 $X_{i,new}$ = normalized value;
 X_i = raw value;
 X_{max} = maximum value of the raw signal;
 X_{min} = minimum value of the raw signal;
 y_i = target values from the RDE tests;

ANN = Artificial Neural Network;
HDV1 = Heavy Duty Vehicle;
LDV2 = Light Duty Vehicle;
LR = Learning Rate;
MAE = Mean Absolute Error;
OBD II = On-Board Diagnostics;
PEMS = Portable Emission Measurement System;
RDE = Real Drive Emission;
ReLU = Rectified Linear Unit.
RMSE = Root Mean Squared Error;
SGD = Stochastics Gradient Descent;

REFERENCES

1. Commission Regulation (EU) 2016/427 (2016)
2. Commission Regulation (EU) 2016/646 (2016)
3. Commission Regulation (EU) 2017/1154 (2017)
4. T. Donato, M. Giovinazzi, "Some Repeatability and Reproducibility Issues in Real Driving Emission Tests" *SAE Technical Paper*, **2018-01-5020** (2018)
5. C. L. Myung, W. Jang, S. Kwon, J. Ko, D. Jin, and S. Park, "Evaluation of the real-time de-nox performance characteristics of a Int-equipped euro-6 diesel passenger car with various vehicle emissions certification cycles," *Energy*, vol. **132**, pp. 356-369 (2017)
6. X. Wang, H. Yin, Y. Ge, L. Yu, Z. Xu, C. Yu, X. Shi, and H. Liu, "On-vehicle emission measurement of a light-duty diesel van at various speeds at high altitude," *Atmospheric Environment*, vol. **81**, pp. 263-269 (2013).

7. L. Qu, M. Li, D. Chen, K. Lu, T. Jin, and X. Xu, "Multivariate analysis between driving condition and vehicle emission for light duty gasoline vehicles during rush hours," *Atmospheric Environment*, vol. **110**, pp. 103-110 (2015)
8. O. V. Lozhkina and V. N. Lozhkin, "Estimation of nitrogen oxides emissions from petrol and diesel passenger cars by means of on-board monitoring: Effect of vehicle speed, vehicle technology, engine type on emission rates," *Transportation Research Part D: Transport and Environment*, vol. **47**, pp. 251-264 (2016)
9. R. Jaikumar, S. S. Nagendra], and R. Sivanandan, "Modal analysis of real-time, real world vehicular exhaust emissions under heterogeneous traffic conditions," *Transportation Research Part D: Transport and Environment*, vol. **54**, pp. 397-409 (2017)
10. D. Lejri, A. Can, N. Schiper, and L. Leclercq, "Accounting for traffic speed dynamics when calculating copert and phem pollutant emissions at the urban scale," *Transportation Research Part D: Transport and Environment*, vol. **63**, pp. 588-603 (2018)
11. M. Fischer, "Transient NO_x estimation using artificial neural networks," *IFAC Proceedings Volumes*, vol. **46**, no. 21, pp. 101-106 (2013).
12. G. Najafi, B. Ghobadian, A. Moosavian, T. Yusaf, R. Mamat, M. Kettner, and W. Azmi, "Svm and anfis for prediction of performance and exhaust emissions of a si engine with gasoline-ethanol blended fuels," *Applied Thermal Engineering*, vol. **95**, pp. 186-203 (2016)
13. H. C. Krijnsen, R. Bakker, W. E. J. van Kooten, H. P. A. Calis, R. P. Verbeek, and C. M. van den Bleek, "Evaluation of fit algorithms for NO_x emission prediction for efficient denox control of transient diesel engine exhaust gas," *Industrial & Engineering Chemistry Research*, vol. **39**, no. 8, pp. 2992-2997 (2000)
14. J. Ma, F. Xu, K. Huang, and R. Huang, "Improvement on the linear and nonlinear auto-regressive model for predicting the NO_x emission of diesel engine," *Neurocomputing*, vol. **207**, pp. 150-164 (2016)
15. I. Arsie, D. Marra, C. Pianese, and M. Sorrentino, "Real-time estimation of engine NO_x emissions via recurrent neural networks," *IFAC Proceedings Volumes*, vol. **43**, no. 7, pp. 228-233 (2010)
16. S. Roy, R. Banerjee, A. K. Das, and P. K. Bose, "Development of an ANN based system identification tool to estimate the performance-emission characteristics of a crdi assisted cng dual fuel diesel engine," *Journal of Natural Gas Science and Engineering*, vol. **21**, pp. 147-158 (2014)
17. S. Uslu and M. B. Celik, "Prediction of engine emissions and performance with artificial neural networks in a single cylinder diesel engine using diethyl ether," *Engineering Science and Technology, an International Journal*, vol. **21**, no. 6, pp. 1194- 201 (2018)
18. M. Canakci, A. N. Ozsezen, E. Arcaklioglu, and A. Erdil, "Prediction of performance and exhaust emissions of a diesel engine fueled with biodiesel

- produced from waste frying palm oil," *Expert Systems with Applications*, vol. **36**, no. 5, pp. 9268-9280 (2009)
19. L. Guzzella and A. Sciarretta, *Vehicle Propulsion Systems: Introduction to Modeling and Optimization*. Springer-Verlag Berlin Heidelberg (2007)
 20. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980 (2015)
 21. A. Langevin and D. Riopel, *Logistics Systems: Design and Optimization*. Springer US, (2005)
 22. M. A. Gelbart, J. Snoek, and R. P. Adams, "Bayesian optimization with unknown constraints," *ArXiv*, vol. abs/1403.5607 (2014)
 23. J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *ArXiv*, vol. abs/1206.2944 (2012)
 24. J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, Red Hook, NY, USA, p. 2546-2554 (2011)
 25. I. Dewancker, M. McCourt, and S. Clark, "Bayesian optimization primer," <https://bit.ly/3d9LiRn>. (2015)